PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-099226

(43)Date of publication of application : 07.04.2000

(51)Int.Cl. G06F 3/00

G06F 17/30

(21)Application number : 11-214490

(22)Date of filing : 29.07.1999

(71)Applicant : XEROX CORP

(72)Inventor : TENEV TICHOMIR G

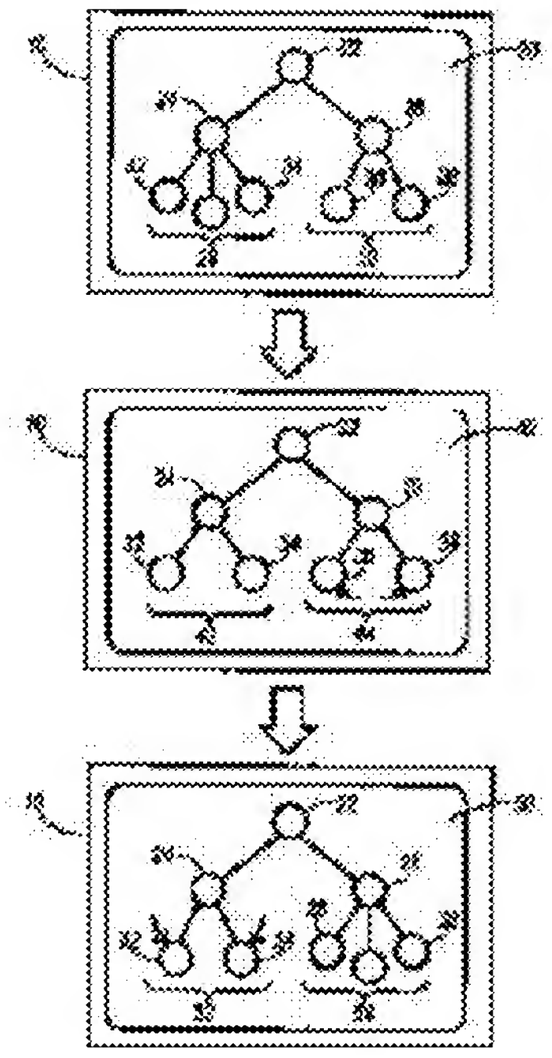
RAO RAMANA B

LAMPING JOHN O

(30)Priority

Priority number : 98 124528 Priority date : 29.07.1998 Priority country : US

(54) METHOD FOR DISPLAYING NODE LINK REPRESENTATION ON DISPLAY



(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method for display in node link structure accompanied by alteration.
SOLUTION: This method comprises a sequence of a 1st step, a final step, and steps including at least one intermediate step; and the 1st representation 20 of a 1st node link structure is displayed in the 1st step and the final representation 50 of a 2nd node link structure is displayed. Common elements of the 1st and 2nd node structures include moving elements represented by features 32, 34, 36, and 38 which are different in position between the 1st representation and final representation and intermediate representation 40 including features representing subsets of common elements including moving elements 32, 34, 36, and 38 individually are displayed in intermediate steps; and the individual subsets are also represented by the features in the 1st and final representations and the features 32, 34, 36, and 38 representing the moving elements have object changelessness through the sequence of the steps.

LEGAL STATUS

[Date of request for examination] 06.07.2006

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's

decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

CLAIMS

[Claim(s)]

[Claim 1] It is the approach of displaying a node-link expression on a display. This approach The 1st step, It consists of the sequence of the step containing the last step and at least one middle step. The 1st step displays the 1st expression showing 1st node-link structure. The last step displays the last expression showing the 2nd node-link structure which was changed by at least one insertion within 1st node-link structure, and at least one deletion and which is a version. The migration element with which it is expressed by the feature in which 1st and 2nd node-link structure has the location where share elements differ in the 1st and the last expression, both including 1 set of share elements is included. A middle step displays the middle expression containing the feature showing the subset of the share element which contains a migration element separately. The approach of displaying a node-link expression on a display that the feature to which each subset is also expressed by the feature in the 1st and the last expression, and expresses a migration element has object permanence through the sequence of a step.

[Translation done.]

* NOTICES *

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the display of the node-link structure which modification produces.

[0002]

[Description of the Prior Art] Journal OBU visual run GEJIZU- and - computing magazine (Journal of Visual Languages and Computing), 1996, the 7th volume, and 33 pages -55 pages, A run ping (Lamping, J.) and RAO (it Rao(s)) "hyperbolic browser of R. : a big hierarchy The focal + context technique ("The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies") for visualizing" The technique for displaying a hierarchy whose number of components increases exponentially is indicated the size of a component decreasing and increasing a radius as a component moves outside. A display is transformed smoothly and a node is changed with a core or a focus. The transition which animated between different views maintains the permanence of an object, and a user helps to adapt himself to change. For quick regeneration, a fringe can be drawn fewer, a line can be drawn instead of an arc, and a text can be omitted during an animation display. U.S. Pat. No. 5,619,632 of run pings (Lamping et al.) is indicating the browser using the similar technique for displaying node-link structure, and has mentioned that a browser may be used for edit of structure.

[0003] Systems - and - KOMPYUTAZU Inn Japan (Systems and Computers in Japan) 1993, the 24th volume, No. 8, 35-page -46-page MATSUURA (Matsuura, T.), Taniguchi (Taniguchi, K.), MASUDA (Masuda, S.) and Nakamura (Nakamura, T.) "the graph editor for the big tree which has browsing and zoom capacity () ["A Graph Editor for Large Trees] with Browsing and Zooming Capabilities"" is indicating the library program which makes possible the direct control of the big tree on a display. DS will be updated

if a node or a subtree is added or deleted. That is, the layout about each node is calculated based on a graph layout algorithm, and each node and an edge are redrawn. If a node is added or deleted continuously, after all modification is made, only one renewal of a display can be performed. A view port depends updating on the mode which is renewal of automatic, the renewal of a part, or renewal of on demand one and to obtain including a global view and canvas. Although the renewal of a part resembles renewal of on demand one, a middle display node is displayed temporarily.

[0004] This invention deals with the problem in the display of the node-link structure of changing.

[0005] Node-link structure may change in some situations. data available to given time amount when structure is large only define a part of structure -- ********** . Although the definition with that the data which define perfect structure have not arrived or very large structure, and perfect structure is held, since memory is not large enough, this may be produced. In such a case, structure partial in memory is held, and it is changed, when additional data arrives or is taken out.

[0006] Another situation that node-link structure changes is the case where structure is the partial version of structure which makes a foundation. For example, a user can require that the specific node or specific link of a type should be chosen, in order for the filtered version to come to hand. It is answered, the node-link structure of omitting the node and link which are not chosen by filter criteria is generated, and a large tooth space is given by a target node and a target link. Or structure may be a tree showing the directed graph (DG) structure of making a foundation, and two or more close links to the node in DG which has a child further exist in directed graph structure. Although the DG node is expressed with one approach by two or more parent nodes which have one of the close links of each in a tree, the child of the DG node is represented only with one of the parent nodes in a tree. A user's input may be answered, the parent node with which a child is represented may change, and a tree structure changes according to it (however, DG does not change).

[0007] Moreover, the foundation structure itself may change. For example, foundation structure may be answered and changed into processes other than the browser which is looking at the foundation structure. Or foundation structure may be answered and changed into a demand of the user through a browser interface.

[0008] The node-link structure of changing by these approaches of arbitration is called "dynamic node-link structure" on these specifications. therefore, this vocabulary changes not only in the basic node-link structure of changing or is static or dynamic, in itself -- the partial version obtained by the approach of being different

from the basic node-link structure as which any are sufficient is also included.
[0009] The display technique and other Prior arts which are stated by the run ping and RAO may be led to the problem accompanying a structural change on display. A layout typically new about a part of structure which changed at least must be made, and then regeneration of it must be carried out. therefore, the expression of the structure which changed when change arose in structure on display using such a technique -- often -- abrupt -- or discontinuous transition exists. In some cases, a layout and regeneration are very late, and it cannot have a dialog on the structure and the effectiveness target from which the user changed. Furthermore, a layout and regeneration do not offer the approach of animation the transition to the old new structure which will raise an understanding of a user's transition from structure. Therefore, many of conventional display techniques only generate a satisfactory result to static node-link structure, and it is not suitable for dynamic node-link structure.
[0010]
[Problem(s) to be Solved by the Invention] However, some of conventional display techniques can animate the transition which extends or contracts a hierarchy list item. However, it is difficult to limit these techniques to an escape or modification of any one contraction, therefore to use widely to all transition.
[0011]
[Means for Solving the Problem] This invention eases the problem in the display of dynamic node-link structure by offering the technique which makes it easy that a user understands transition of the larger range. Each technique displays an expressional sequence, the first expression expresses 1st node-link structure, and the last expression expresses the 2nd node-link structure which is change-bar JON of 1st node-link structure. The element shared according to the structure of the beginning and the last contains the migration element with which it is expressed by the feature (feature) which has the location in which the first expression differs from the last expression. This technique displays at least one middle expression between the first expression and the last expression. Each expression in this sequence contains the feature showing the subset of the share element containing a migration element. The feature showing a migration element has object permanence through the sequence of a step.
[0012] In one mode of this invention, 2nd node-link structure is change-bar JON accompanied by at least one insertion in 1st node-link structure, and at least one deletion. For example, an element is deleted from one location and may be inserted in

another location. Therefore, the technique called "deletion and insertion" on these specifications in which this mode was followed may be used in order to move an element, or in order to make [rather than] a complicated change including two or more deletion and insertion.

[0013] New deletion and an insertion technique display the sequence of an image to which the field of the element under deletion or insertion carries out change of reduction or one of increases in each of a sequential image, respectively. Compensation field change arises to a neighboring element at coincidence. When field change is chosen appropriately, the sequence of an image approaches the element which a neighboring element is deleting in deletion, and produces the impression to which the element under insertion pushes aside a neighboring element by its side in insertion.

[0014] In each case, the sequence of an image can also offer a suitable motion of the element under deletion or insertion. For example, in the deletion of an element which has a wedge-shaped field, before the element under deletion can come out from the edge which the wedge opened, or can be shrunken along with the wedge or its wedge of the begins to shrink, it can disappear.

[0015] Another mode of this invention is based on discovery of the technique which animates transition on a two-dimensional display. According to this technique, two migration elements follow the pass which is not an parallel straight line. For example, an element may move along with curvilinear pass and may move along with the independent pass which is not parallel.

[0016] Still more nearly another mode of this invention is based on discovery of the technique which animates modification according to individual of node-link structure in parallel. According to this technique, the signal which requires modification according to at least two individuals is received, and the modification data in which modification demanded based on this signal is shown come to hand. Next, the 2nd DS which defines change-bar JON of node-link structure comes to hand using the 1st DS which defines modification data and node-link structure, and an expression can be displayed according to it.

[0017] Since this new technique makes it easy that a user understands the transition within dynamic node-link structure, it is useful. Furthermore, this new technique does not carry out count which time amount required for the layout of perfect node-link structure and activation of a display requires, but can use for and carry out increment layout and regeneration instead. Consequently, a user can have a dialog with dynamic node-link structure more effectively.

[0018]
[Embodiment of the Invention] The vocabulary which the following notional designs are helpful when you will understand the large range of this invention, if it reads with the notional design stated by U.S. Pat. No. 5,590,250 used for this specification as reference and No. 5,619,632, and is defined below has the semantics shown through this application containing a claim.
[0019] "Node-link structure" is the structure where each link is related to two or more nodes including an item distinguishable to a node and a link. A "graph" is the node-link structure where each link is related to two nodes. A "directed graph" is a graph whose node of another side the direction of [between the nodes related to each link] is shown, one node is the source or a "FUROMU node" of a link, and is the destination or a "two node" of a link. A "non-ring type directed graph" is a directed graph which does not offer the pass which returns from the node of arbitration to itself, when a link is followed in those shown directions. A "tree" is a non-ring type directed graph which has only one root node which offers only one pass which starts in a root node and is led to a non-root node about the non-root node of the arbitration in a tree when a link is followed in the shown direction.
[0020] The "element" of node-link structure is the node and link of node-link structure.
[0021] "A node-link expression" is an expression of node-link structure. For example, a link feature can express a link and a node feature can express a node.
[0022] When a certain feature showing an element has the spatial range of each dimension of the display displayed, the feature "has a field." For example, on a two-dimensional display, when a feature is prolonged into the dimension of both displays, the feature has a field.
[0023] The "neighboring feature" of the feature showing a certain element in a node-link expression is close with a certain feature including the feature showing other elements from the median of the distance between other features to which a neighboring feature expresses the element of a certain feature and others.
[0024] When all a certain features showing the element in the expression in a sequence seem to be the same feature, the feature showing the element of node-link structure "has object permanence" through the sequence of an expression of node-link structure. For example, the feature in two continuous expressions of the arbitration in a sequence may be similar enough, and may be close to time amount and a space target enough so that they may be visible to the same feature. the feature which has the permanence of an object -- "pass -- following -- " -- it seems to

move.
[0025] An "animation loop formation" is each repetitive actuation which displays an image repeatedly, and looks like continuation of the feature in each precedence image to which a degree is followed [repeatedly] so that according to object permanence in the feature in each image. An "animation cycle" is once [of an animation loop formation] repeatedly.
[0026] The element "is extended" when a certain element has at least one child in a node-link expression. ["which has "descendant within a node-link expression, or] When a certain element does not have a child at all in an expression, the element "does not have a descendant." The element "is contracted" when it does not have a descendant at all in an expression, in spite of basing a certain element on the node-link structure of making a foundation.
[0027] On these specifications, the vocabulary a "navigation signal" means the signal which shows that the user has an interest higher than other parts in the part with node-link structure. For example, an "extended signal" shows the demand which displays the graphic representation to which the expression of an element with a graph is extended, and, on the other hand, a "contraction signal" shows the demand which displays the graphic representation by which the expression of an element with a graph is contracted. It is possible by the point and click actuation which require that it should depend especially or the feature for which it chooses a bookmark etc. including the demand whose examples of other display a part of node-link structure in a specific location, and to which it pointed should be moved to a focal core.
[0028] Like insertion of one or more elements or deletion, like the migration which can be carried out with the combination of insertion and deletion, or copy actuation, when a certain signal requires modification of one or more elements in structure, as for the signal, "modification is required" within node-link structure. An escape and a contraction signal are the examples of the signal which requires modification.
[0029] When two change may be respectively made independently of another side, two modification "is individual."
[0030] In drawing 1 , a display 10 begins with the 1st expression 20 of 1st node-link structure, and shows the sequence of the node-link expression ended with the last expression 50 of the 2nd node-link structure which is change-bar JON which performed both deletion which may be produced from the signal which requires modification according to two individuals of 1st node-link structure, and insertion. A display 10 displays the middle expression 40 among expressions 20 and 50.
[0031] 1st node-link structure has the 1st level node expressed by the feature 22,

and the 1st level node has two children who are the 2nd level nodes expressed by features 24 and 26. The child to whom it is expressed by the feature 24 is at the top of a branch 28, and a branch 28 contains all descendants including three children who are the 3rd level nodes. Similarly the child to whom it is expressed by the feature 26 is at the top of a branch 30, and a branch 30 contains all descendants including two children who are the 3rd level nodes.

[0032] the 2nd node-link structure shown by the last expression 50 -- many elements -- 1st node-link structure -- sharing -- **** -- the subset -- expressions 20, 40, and 50 -- it is expressed by the feature of all. A share element has the location by which each was stabilized through expressions 20, 40, and 50 including the node expressed by features 22, 24, and 26 and the link to which they are connected. The node to which a share element is similarly expressed by the features 32 and 34 in a branch 28, The node expressed by the features 36 and 38 in a branch 30, Each feature showing one of these nodes and links including the feature showing the link to nodes 32, 34, 36, and 38 It has a different location from the inside of the 1st expression 20 within the last expression 50, therefore these nodes and links are called a "migration element" on these specifications.

[0033] In the example shown in drawing 1 , change of the location of the feature showing a migration element is produced for reasons of having been changed at two points, in order that 1st node-link structure may acquire 2nd node-link structure. In order to delete one of the 3rd level nodes expressed within the branch 28 in order to obtain a branch 52 and to obtain a branch 54, the additional 3rd level node is added to the branch 30. It is not concerned with change of the location produced as a result in the modification list between 1st and 2nd node-link structures, but the feature showing a migration element has object permanence through an expressional sequence. Nodes 36 and 38 seem to follow and leave curvilinear pass, and, on the other hand, nodes 32 and 34 seem to follow the curvilinear pass which goes mutually so that it may be shown by the arrow head in expression 40 and 50. Therefore, these nodes seem to follow the pass which is not an parallel straight line.

[0034] Although the object permanence which lets a sequence pass can be produced by various approaches, the technique shown in drawing 1 shows the example which produces object permanence by displaying a similar feature at the small interval of space and time amount. While being displayed continuously in the same location quickly, it is substantially the same, and each of features 22, 24, and 26 and the feature which expresses the link between them to a list can be continued, and secures object permanence. While each of the feature showing the link between them is also

quickly displayed on features 32, 34, 36, and 38 and a list continuously in the location displaced slightly, it is possible for it to be substantially the same and to continue. The variation rate of the location of features 32 and 34 brings them close mutually, and compensates them with deletion of the 3rd level node which is the child of a feature 24. That is, features 32 and 34 are more nearly mutually [than the inside of the branch 42 by which deletion was performed to the inside of the branch 28 before deletion, and the beginning] close within the branch 52 from which the node was deleted so that it may illustrate. in order that the variation rate of features 36 and 38 may compensate insertion of the 3rd level node which is the child of a feature 26 -- features 36 and 38 -- alienation -- it moves. That is, features 36 and 38 are further estranged rather than the inside of a branch 30 within a branch 44, and the 3rd level node is inserted within the branch 54.

[0035] In drawing 2 , processing of the processing box 100 is started by displaying the 1st expression. The 1st expression contains the feature which expresses 1st node-link structure and expresses the subset of a 2nd node-link structure and shared element. A subset contains a migration element like the node expressed with one of the features 32, 34, 36, and 38 of drawing 1 .

[0036] Next, processing of the processing box 102 displays the middle expression containing the feature which similarly expresses the subset of a share element. As shown by the dotted line, processing in the processing box 102 may be performed once or more, in order to display a series of two or more middle expressions. Within each middle expression, each feature showing a migration element has object permanence with the feature showing the migration element in the precedence expression which is the 1st expression or a middle expression to precede, and is obtained.

[0037] Finally, processing of the processing box 104 displays the last expression showing the 2nd node-link structure which is change-bar JON of 1st node-link structure. Like the processing box 102, it has object permanence with the feature showing the migration element in the middle expression which the feature showing a migration element is not concerned with change of the location from the 1st expression to that last table current, but precedes the last expression including the feature showing the subset of a share element.

[0038] The equipment 150 of drawing 3 contains the processor 152 connected in order to supply the data which define an image as a display 156, while receiving the data in which a user signal is shown from the user input circuit 154. The processor 152 is connected also in order for the 2nd structure to access the node-link data 158

which define the 1st and 2nd node-link structure which is change-bar JON of the 1st structure. The processor 152 is connected also in order to receive the instruction data 160 in which an instruction is shown through memory 164, storage access equipment 166, or the instruction input circuit 162 that can supply the instruction received from connection with a network 168.

[0039] In activation of the instruction shown by depending instruction data 160, a processor 152 displays the 1st expression of the 1st node-link structure containing the feature which expresses the subset of a 2nd node-link structure and shared element to a display 156. The subset of a share element contains a migration element as mentioned above. Next, similarly a processor 152 displays on a display 156 at least one middle expression containing the feature to which each expresses the subset of a share element. That is, the feature showing a migration element has object permanence with the feature showing the same share element in a precedence expression. Finally, the feature which a processor 152 makes display the last expression of the 2nd node-link structure which contains in a display 156 the feature which similarly expresses the subset of a share element, and expresses a migration element has object permanence with the feature showing the migration element in the middle expression to precede.

[0040] As mentioned above, drawing 3 shows the three possible sources 164 which can receive the data which the instruction input circuit 162 shows an instruction, i.e., memory, storage access equipment 166, and a network 168.

[0041] the conventional memory of the arbitration in equipment 150 in which memory 164 contains random access memory (RAM) or read-only memory (ROM) -- or you may be the circumference or the remote memory apparatus of a class of arbitration.

[0042] Storage access equipment 166 may be the drive for accessing the storage 170 which are other suitable media of the arbitration for storing or more 1 sets of one tapes, a diskette or a magnetic medium like a floppy disk, an optical medium like or more 1 sets of one CD-ROMs, or data, and is obtained, other suitable equipments, or a circuit. A storage 170 may be some equipments 150, a part of server, other circumferences, a remote memory apparatus, or a software product. In each in these cases, a storage 170 is one product which can be used into equipment 150. It is able to arrange a data unit, and for storage access equipment 166 to access a data unit, and to enable it to supply them by the sequence through the instruction input circuit 162 at a processor 152 on a storage 170. If a data unit is supplied by the sequence, it will form the instruction data 160 to which it is indicated that it is illustrated by instruction.

[0043] A network 168 can supply the instruction data 160 received from equipment 180. The processor 182 in equipment 180 crosses a network 168, and can establish connection with a processor 152 through the network connection circuit 184 and the instruction input circuit 162. Which processor may start connection and connection may be established with the suitable protocol of arbitration. Next, since a processor 182 accesses the instruction data stored in memory 186, crosses a network 168 and can transmit instruction data to a processor 152, a processor 152 can receive the instruction data 160 from a network 168. Next, it is stored in memory 164 or somewhere by the processor 152, and the instruction data 160 can be performed.
[0044] Drawing 4 shows how modification according to individual can be processed in parallel.
[0045] Processing of the processing box 190 receives the signal which requires modification according to at least two individuals in 1st node-link structure. For example, a signal may also include two different signals which requires modification according to individual.
[0046] Based on this signal, processing of the processing box 192 receives the modification data in which modification demanded by the signal is shown. Modification data may be stored by the suitable DS according to use.
[0047] Next, processing of the processing box 194 receives the 2nd DS which defines 2nd node-link structure using the 1st DS which defines modification data and 1st node-link structure. 2nd node-link structure is the version by which modification required of 1st node-link structure was performed.
[0048] Finally, processing of the processing box 196 displays the expression of 1st and 2nd node-link structure with animation. This processing can include the processing mentioned above with regards to drawing 2 .
[0049] The general feature mentioned above may be carried out on various equipments by many approaches for displaying a node-link expression. The example described below runs the 32-bit version of Microsoft Windows (Microsoft Windows), and is carried out on the system based on PC which performs the code compiled from the C++ language source code.
[0050] In drawing 5 , a system 200 contains the PC processor 202 connected to the mouse 208 and keyboard 206 for supplying a signal from the display 204 for displaying an image, and a user. The PC processor 202 is connected also so that memory 210 and a client 212 can be accessed. Memory 210 can contain program memory 214 and data memory 216 as illustrated. the combination of the routine and data with which the client 212 was stored in memory 210 -- or it is the source of the information

about a directed graph which may be independent of memory 210 so that it may be shown. For example, a processor 202 may communicate with a client 212 through a network.

[0051] The routine stored in program memory 214 can carry out grouping to some functions. The GURAFA routine 220 generates and changes the DS showing the directed graph defined using the information from a client 212. The walker routine 222 answers a navigation signal and other user signals from a keyboard 206 and a mouse 208 by information coming to hand from directed graph DS. The painter routine 224 provides a display 204 with a signal, and displays the expression of directed graph DS on it. The mathematics routine 226 can be called in order to obtain the location of the element of the directed graph in layout space.

[0052] Next, data memory 216 holds the DS accessed by the processor 202 during activation of the routine in program memory 214. As mentioned above, directed graph DS 230 may be accessed by the walker routine 222 and the painter routine 224 while it may be generated and changed by the GURAFA routine 220.

[0053] It is linked in directed graph DS 230, and it can obtain or the node location data 232 which may be contained can include the location of the node in **** space like a hyperbolic plane, and rendering space like a two-dimensional unit disk (unit circle). The node location data 232 may be accessed by the routine in program memory 214.

[0054] The routine in program memory 214 can also access various coarse DS 234. DS 234 may also include the preliminary data structure for mapping to Link ID from one pair of nodes ID carried out as for example, a standard heap. That is, this preliminary data structure enables the retrieval and insertion of Link ID within fixed expected time.

[0055] Drawing 6 shows how an event can be answered when the system of drawing 5 displays the expression of a graph.

[0056] With the processing box 300, a client 212 is started by loading the initial set of an element to memory, for example through the call of node generation while an initiation graph comes to hand. An extended flag defines the tree in the initial set of an element. A client 212 also performs the suitable call to the routine in memory 214, in order to perform all the things for which the version by which paint was carried out by arranging a tree to a hyperbolic plane, carrying out the paint of the tree which mapped the root node from the hyperbolic plane at the core of a disk to the unit disk, and was mapped in the tree, and carrying out the swapping of the double buffers is displayed on a display 204 with the processing box 300.

[0057] With the processing box 302, a client 212 receives the event related to a graph.

An event may be produced from a navigation signal, an edit signal, or the signal of other types of a user. Or an event may be received from other sources of the interior of a system 200, or one of the exteriors. Anyway, an event can take the gestalt of the call from other instructions executed by the processor 202 from one of the routines in [out of a client 212] memory 214. The event by which a single string was received may be held in a queue so that the processing box 302 may include the ejection of the event from a queue.

[0058] A client 212 starts a suitable response by answering the event received with the processing box 302 by performing one or more calls to the routine in memory 214. Since a response is based on the type of an event so that it may be shown by the decision box 304, a branch is chosen based on an event.

[0059] An event may be a non-animation event like an orientation conversion event, a stretch event, or a drag event. An orientation event may be produced when a user shows new orientation to a root node. A stretch event may be produced when a new stretch factor is shown to the expression with which the user is displayed. A drag event may be produced, when a user chooses the location in an expression by mouse down (have pressed mouse button) click etc. and demands migration of the location with suitable gesture or other suitable signals.

[0060] With the processing box 310, a client 212 starts the response to a non-animation event by the information on arbitration required for the response to an event coming to hand. About an orientation event, the information which came to hand with the processing box 310 can include new orientation. About a stretch event, the information which came to hand with the processing box 310 can contain a new stretch factor.

[0061] About a drag event, complexity of acquisition of the information on the processing box 310 increases a little. A client 212 may receive the information about the motion which could receive the node identifier (node ID) of the nearest node from the selected position, and was demanded. these informational items are indicated with regards to the column 71-72 and drawing 14 of U.S. Pat. No. 5,590,250 which are used for this specification as reference -- " -- the nearest node may come to hand by the almost same approach as it is explained by the function (function) retrieval."

[0062] A client 212 can be rounded off with the suitable call to the walker routine 222 and the painter routine 224 for a layout, mapping, and paint if a client 212 receives required information with the processing box 310. About an orientation event, a root node must be arranged for new orientation. About a stretch event or a drag event, a layout is unnecessary. However, about a stretch event, the call to the walker routine

222 must contain the new stretch factor for using for mapping. Similarly, about a drag event, the call to the walker routine 222 must include the location of a degree which met the node ID of the nearest node for using for mapping, and the pass of a motion.
[0063] With the processing box 312, first, the walker routine 222 may perform the required layout of arbitration in a hyperbolic plane, and may arrange the edit under hold of the arbitration of a tree. Next, with the processing box 314, the walker routine 222 may start an initiation node in a starting position, and may map a tree on a unit disk. For example, a drag event may be answered, an initiation node may be nearest node identified with the processing box 310, and a starting position may be a location of a degree which met the pass of a motion. The initiation node and starting position which were before used for mapping may be answered and used for orientation or a stretch event.
[0064] If a tree is mapped, in order to carry out the paint of the mapped tree to a display buffer with the processing box 316, the painter routine 224 can be called. The painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation among paint. Each edit can carry out marking by [which set up a flag] depending especially or storing other suitable data. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph will be supplied.
[0065] As mentioned above, current is carried out for these events as a non-animation event. Answering an orientation event, an expression circles in the surroundings of the node which is in new orientation typically at the focus of a viewing area. Similarly, a stretch event is answered, and an expression is extended or contracted around the node which is in radial typically at a focus. Answering a drag event, an expression moves at the rate determined by the input signal. However, a client 212 may supply the response animated in demanded modification by depending, changing into the sequence of a small event, being the processing box 310 and emitting a series of equivalent calls to an event with one call smaller than each one to an orientation event, a stretch event, or a drag event.
[0066] Drawing 6 also shows the response to the event of two different types which always animates and are treated by the present operation. The 1st type is a bookmark or a click event, it is answered, the location of one node is moved into an animation sequence, and other elements move it according to a motion of the one node. Although one node has been stabilized in the animation sequence to which the 2nd type is an insert/delete event, it is answered, a certain element is contracted, and another element is extended, other elements move to compensate for contraction and

an escape.

[0067] A bookmark or a click event may be produced, when the item in the set of a user of a menu or other bookmarks is chosen, or when the location in an expression is chosen by mouse down rise (mouse button is pressed and released) click. Answering this type of event, a client 212 receives the destination location in Node ID and a unit disk. In the case of the bookmark event, it is stored before, and it can search Node ID and a destination location from memory. in the case of the click event, the client 212 is indicated with regards to the column 71-72 of U.S. Pat. No. 5,590,250 used for this specification as reference -- " -- it may be the almost same approach as the function [node / nearest] (function) retrieval", and the node ID of the nearest node may come to hand from a selected position, and the destination may be a default location like the core of a unit disk.

[0068] With the processing box 320, a client 212 may call the walker routine 222 in Node ID and a destination location. The walker routine 222 can answer by executing the animation loop formation for displaying the sequence of the expression which a node moves to a destination location from a front location. With the processing box 320, the walker routine 222 is started by setting up a pair of sequence of the node/location where each includes the location in Node ID and a unit disk. A location can come to hand by repeating the total translation (migration way) from a location to a front destination location coming to hand, and then the n-th root of the total translation coming to hand, and constituting with the present translation as indicated in the boxes 470, 472, and 482 of drawing 12 of U.S. Pat. No. 5,619,632. A pair of number of a node/locations can be done greatly enough, in order to secure the smooth animation from a location to a front destination location, while the feature which expresses the element of structure during an animation display maintains object permanence. A location may come to hand by choosing a suitable number of points from a front location instead of the approach of the n-th root along with the arc appropriately chosen in the hyperbolic plane in the destination location. An arc may be chosen so that between the straight line which may look unnatural, and the arcs which the node probably took by the approach of the n-th root and which may need too much animation number of steps for looking smoothly may be compromised. The number of points may be chosen so that satisfactory animation may be secured.

[0069] Next, the walker routine 222 repeats an animation loop formation about each set of the node/location in a sequence, as shown by the decision box 322. With the processing box 324, the walker routine 222 may arrange the edit under hold of the arbitration of a tree to a hyperbolic plane first, as mentioned above with regards to the

processing box 312. Next, with the processing box 326, the node and location from the pair of a node/location of a degree may be started as an initiation node and a starting position on a unit disk, and the walker routine 222 may map a tree on it, as mentioned above with regards to the processing box 314.

[0070] If a tree is mapped, in order to carry out the paint of the tree mapped with the processing box 328 to a display buffer, the painter routine 224 can be called. Among paint, the painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation, as mentioned above in relation to the processing box 316. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph can be offered.

[0071] If marking of the new edit is carried out by the painter routine 224 with the processing box 328, new edit will be arranged with the processing box 324 during the next iteration. Consequently, the sequence by which the expression was animated shows dynamic node-link structure rather than shows static node-link structure like U.S. Pat. No. 5,629,632. However, since fundamentally adds the feature which expresses a new node along with an expressional periphery when an expression changes from a front location to a destination location, this edit works. Consequently, the consciousness of the object permanence about the feature showing other elements blocks, or the added feature does not reduce it.

[0072] An insert/delete event may be produced, when a user demands an escape or contraction of a node, or when modification of everything but some of graphs or trees is required. An insert/delete event may offer the device for automatic modification of the graph which it may be received [graph] also with the gestalt of a call, therefore does not make control by people's hand concurrent, or a tree. Furthermore, an insert/delete event may be produced from the signal which requires modification according to two or more individuals which has data in which modification which suspended the call to the animation of all modification and was obtained is shown. One insert/delete event may also include both deletion and insertion.

[0073] This type of event is answered, and it is the processing box 330, and a client 212 can perform the suitable call to the routine in memory 214 first, in order to determine whether modification of the demanded graph or tree is permissible.

[0074] When modification of the demanded graph or tree can be permitted, according to it, a client 212 calls the routine in memory 214 if needed, and can change a graph or a tree. In the process into which it changes, marking of each element called "the element by which it was influenced" is carried out by a setup of a flag, or storing of

other suitable data in these specifications which insert and delete or may be changed by modification. Since the field assigned to the parents may change when a certain node is chosen by the extended signal or the contraction signal, it is the node by which the parents were also influenced. Only the node inserted or deleted is influenced about most of other insert/delete events. A certain node can choose a client 212 as a stabilization node held also during the animation display of modification in the location in front of the node.

[0075] In many cases, the initiation node used for mapping can be chosen as a stabilization node in front, and the node can be held to a front starting position. In other cases, selection of a different stabilization node is sometimes desirable for a client 212. For example, the node currently extended may be chosen as a stabilization node held in a current location, therefore the node serves as a new starting position. Therefore, unless the stabilization node from which a client 212 differs is chosen, a front initiation node and a front starting position are usually maintained. However, when the front initiation node is deleted, other nodes must be chosen as a default stability node changed by the client 212.

[0076] When deletion is performed, the walker routine 222 can be called by the node ID of the nearest ancestor that remains even after deleting in the node under deletion, and the tree under mapping. This ancestor can find it by what (walk) is patrolled upward until it reaches the ancestor who is not deleted from the node under deletion by the present insert/delete event.

[0077] This call is answered and, as for the walker routine 222, the node under deletion can test whether it is a front initiation node. When that is right, the identified ancestor can be chosen as an initiation node in order to permute by the node under deletion. It is mapped by the location where the ancestor is displayed recently, and when it is an usable location, the location can be chosen as a starting position. When it is mapped by the unusable location since the location which is not displayed, or the element besides the present are mapped or an ancestor was not mapped recently, a starting position can be set as the core of a unit disk.

[0078] Also in the processing box 330, a client 212 may call the walker routine 222 in ID and the location of a stabilization node. The node from which the walker routine 222 was deleted first is contracted in a front location, the node inserted in the degree is extended in a new location, and it can answer in the meantime by executing the animation loop formation which displays the expressional sequence that a stabilization node is held in a front location, all the time. Since a stabilization node is mapped by the location or the unusable location which is not displayed or it was not mapped

recently, when it cannot hold in a front location, it is in the condition which has a front initiation node in a front starting position, and after the deleted node is contracted, a stabilization node can move to the location, consequently the sudden motion between contraction and an escape arises. The walker routine 222 is started by setting up the sequence of weight which governs the rate from which the field assigned to each influenced node during contraction and an escape changes. Weight is classified for the increment small enough, in order to maintain object permanence animation on display.

[0079] Next, the walker routine 222 repeats an animation loop formation about each weight in a sequence, as shown by the decision box 332. With the processing box 334, the walker routine 222 may arrange the edit under hold of the arbitration of the node and tree which were influenced first in the hyperbolic plane using the weight of iteration. Next, in the processing box 336, the walker routine 222 may start and map a tree in a stabilization node and a location on a unit disk, as mentioned above with regards to the processing box 314.

[0080] If a tree is mapped, in order to carry out the paint of the mapped tree to a display buffer with the processing box 338, the painter routine 224 can be called. Among paint, the painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation, as mentioned above with regards to the processing boxes 316 and 328. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph can be offered.

[0081] Even if the edit under hold exists and it does not carry out, the expression of the dynamic node-link structure by deletion and/or insertion is generated repeatedly [of a single string of the animation loop formation started with the decision box 332]. Furthermore, the influenced element moves to a new location from the location before deletion and insertion. This technique was well realized, in order to produce the object permanence between these motions.

[0082] After the expression was supplied with the processing box 316, or after an animation sequence is completed by the decision box 322 or 332, another event is receivable with the processing box 302 so that it may be shown by the circle displayed as "A" of drawing 6 .

[0083] Drawing 7 shows how a layout can perform first with the processing box 300 of drawing 6 . Drawing 8 is the processing boxes 312, 324, and 334, and shows how the layout of the changed node-link structure can perform.

[0084] As shown by the processing box 350, the walker routine 222 starts the first layout by using in order for a root node ID to come to hand and to access the data

related to the root node within directed graph DS 232. With the processing box 352, the walker routine 222 arranges a root node by performing the call to the mathematics routine 226 by include-angle width of face. This include angle is good at the suitable include angle of the arbitration which generates a desirable result. When include-angle 2pi and pi/2 were used well and they used 2pi, it was suitable for the main layout style, and when pi/2 was used, it was suitable for the right or a left layout style a top and the bottom. An interface may be offered, in order to obtain a desirable result and to correct this include angle.

[0085] By response, the mathematics routine 226 arranges a root node at the zero of a unit circle, i.e., a coordinate, (0 0) for upward orientation at an angle of the one half of a coordinate (0 1) and include-angle width of face. Next, the walker routine 222 pushes a root node ID in front of a queue with the processing box 354.

[0086] In the remainder of drawing 7 , as shown by the decision box 360, the walker routine 222 scans repeatedly 1 set of elements of the tree defined according to directed graph DS 232 until a queue becomes empty. It is started by repeatedly [each] Node ID coming to hand after a queue by using in order to access at the data related to the node from which it was discriminated within directed graph DS 232.

[0087] It is determined [each] whether, repeatedly, the node already patrolled the test in the decision box 370 by this scan. When not going round, the walker routine 222 carries out marking of the gone round round node with the processing box 372. So that two or more children's node ID comes to hand, the non-extending leaf of two or more children, i.e., a leaf node without the extended close link, may be identified, the number of the children who exist in a visible tree may come to hand so that it may explain below, and it may similarly explain below In order for the include angle about two or more children in a visible tree and the array of a radius to come to hand, the mathematics routine 226 is called by several Ns of the child of the node which exists in a visible tree.

[0088] About processing in the processing box 372, several Ns of the child in the visible tree which is one type of neighboring relational data are calculable with either of two kinds of approaches. That is, when a current scan is a part of sequence of a node addition step, N is equal to the number of the children before a scan, and a child's added number of sum totals. When a current scan is a part of sequence of a node removal step, N is only equal to the number of the children before a scan.

[0089] With the processing box 372, the array of an include angle and a radius can come to hand by various approaches. At one successful example, it is set as the value of 0.7 and, on the other hand, each radius is each include angle. (N*pi) (/18) It is set as

the smaller one with pi. Therefore, the include angle of a node is influenced about N< 18 by the number of the children of the node which exists in a visible tree.

[0090] Next, with the processing box 374, the walker routine 222 calls the mathematics routine 226, in order to arrange a child.

[0091] In case the processing box 374 is carried out, two general principles of a layout are applied. The alienation between nodes and an include angle are determined as the 1st only based on the information about a neighboring element, i.e., the neighboring relational data, in a tree. The layout information received about each node by the 2nd is the approach which the location of a certain node and all its children can move by the small change within DS, and shows the relative position of the parents of node opposite **.

[0092] The general strategy which can follow is that start at a child's radius and include angle from the processing box 372, the approximation distance which a child needs comes to hand, the distance from parents comes to hand using approximation distance, and a more exact distance about a child comes to hand using the distance from parents, next a further more exact distance comes to hand using a more [in optional] exact distance etc.

[0093] When a child has the radius R and include angle theta from the processing box 372 according to a general strategy, the approximation distance D1 and D2 can be calculated as sinh (R) and tan (theta/4), respectively. When N is the number of the children under layout activation and the sum total is performed from i= 1 to N-1 The pair of each child who approached A larger distance of each sum total of D1 and D2, Namely, DT=sigma (when divided by max(D1(i)+D1 (i+1), D2(i)+D2 (i+1))+max(D1 (1), D2 (1))+max (D1 (N), D2 (N))) The total distance DT about all children can be acquired using D1 and D2. When the parents have the usable include angle omega, the distance DP from parents can be calculated as asinh (DT/omega). Next, a child can be stationed the include angle between the children proportional to division of a child in accordance with the periphery of the circle of the radius DP centering on parents.

[0094] Next, a more exact distance about a child can be acquired as follows using DP. D1 -- '=sinh(DP) asin (sinh(R)/sinh (DP)), D2' -- even parents' more exact distance DP' can be obtained until it reaches the accuracy of desired level like an above-mentioned approach using =2sinh(DP) atan (tan (theta/4)/eDP) next D1', and D2'. At the time, each child's orientation is calculable as a gap of the include angle from parents' orientation.

[0095] Please note that an above-mentioned distance is expressed with actual measuring in a hyperbolic plane. The distance D in a hyperbolic plane corresponds to

the vector which begins at the zero in a unit circle and progresses the distance of tanh (D/2).

[0096] As mentioned above with regards to the processing box 372, a general strategy is carried out in this way, and receives the layout information only based on the neighboring relation information about a sibling including the information about a node, its parents, and the child of the sibling in a visible tree. A general strategy receives the layout information which shows the distance from a child to the parents, and the include angle showing the difference in the orientation of a child and its parents.

[0097] The general strategy was carried out as software using the first distance acquired without carrying out the attempt which acquires a more exact distance by the approach mentioned above with regards to the general strategy, although two repetitive loop formations were carried out about all children. The 1st loop formation receives the separation between the adjoining children, and the "slice size" about each child, is saved temporarily and also receives the sum total of separation. Next, the distance to parents is acquired using this information. Next, the 2nd loop formation receives and saves each child's relative orientation and relative field.

[0098] In the example as this software, when a child has the radius R and include angle theta from the processing box 372, as mentioned above with regards to the general strategy, distance D1 and D2 is calculated. D1 and D2 about each child are added to D1 and D2 about the former child, and S1 and S2 are obtained. The sum total ST of separation is just increased to the child of the beginning and the last by a child's maximum of S1 and S2 except for **, and it is increased to the child's maximum of D1 and D2 by ST about the child of the beginning and the last.

[0099] When S1 of a child is larger than S2 of the child, while S1 is saved as separation of a child, D1 is first saved as a child's slice size, and about the child of the 2nd henceforth, it is adjusted so that the former child's slice size may serve as the minimum value of the slice size and S1 of a child in front of the child. On the contrary, when S1 of a child is not larger than S2 of the child, while S2 is saved as separation of a child, D2 is first saved as size of a child's slice, and about the child of the 2nd henceforth, it is adjusted so that the former child's slice size may serve as the minimum value of the slice size and S2 of a child in front of the child. However, the last child's slice size is adjusted so that it may become the minimum value of the slice size and a child's maximum of D1 and D2 in front of the child, and the 1st repetitive loop formation completes it.

[0100] Next, the distance DP from parents is calculable using parents' include angle omega as tanh (asinh (ST / 2omega)/2) or the larger one of 0.5. DP is saved as some

data relevant to a parent node.

[0101] About each child, the 2nd repetitive loop formation is started by calculating the include angle (S/ST) of 2omega, when S is the separation saved about the child. The include angle (S/ST) of 2omega is applied to the running total started by -2omega. A running total is saved with other data relevant to a child.

[0102] Next, the mathematics routine 226 can calculate the new include angle about a child by calling a function similar to the "interior angle (inside-angle)" function of the columns 67 and 68 of U.S. Pat. No. 5,590,250. On these specifications of these, the function called "an inside angle type (InsideAngle)" is started at an angle of the one half of the distance ("dist") and the wedge which were moved into the wedge. An inside angle type takes the smaller one of an initiation include angle and (pi-epsilon) in case epsilon may have a very small value like 0.0001 as an include angle which can be operated, and avoids the problem in count of an inverse tangent. An inside angle type receives the deformation which moves the point in the coordinate on a unit circle (dist, 0) to a zero. Next, an inside angle type applies this conversion to the complex-locus label of the point on the intersection of the periphery of a unit circle, and the radiation which is emitted from a zero and whose include angle with a horizontal position is an include angle which can be operated. An inside angle type returns the include angle from the horizontal position of the radiation which passes along the point changed from the zero as an include angle of a result.

[0103] In order to obtain a child's include angle, an inside angle type is called at the include angle calculated by carrying out the multiplication of distance DP and the slice size of the child from the 1st iteration by 2omega / ST. The include angle returned by the inside angle type is measured with pi/2, and a child's include angle is the smaller one of these two.

[0104] Before saving a child's new include angle, the mathematics routine 226 saves the include angle in front of a child. When the absolute value of the difference of an old include angle and a new include angle exceeds the minimum value, the mathematics routine 226 also saves the data in which it is shown that a layout should continue so that it may state below.

[0105] the last -- the 2nd iteration loop formation -- "space usable (room-available)" of the columns 67 and 68 of U.S. Pat. No. 5,590,250 -- a child's field or side tooth space is obtained by calling a function similar to a function (function). On these specifications, this function called "available [room available (RoomAvailable)]" is started at an angle of [phi] the one half of the distance D and the wedge which were moved into the wedge. room available -- first -- a ratio (1-D2) -- by obtaining /2D, in

order to acquire the initial distance S next, the distance to the edge of a wedge calculated by breaking a ratio by sinphi is returned. Next, room available returns distance (S2-1) (1 / 2-S). In order to obtain a child's field, room available is called at the same distance and the same include angle as having been used for the call of an above-mentioned inside angle type. The distance returned with room available is saved as a scale of a child's field.

[0106] Although the example as above-mentioned software can save additional data, the example as software is based on discovery that only the data of a dyadic eye need to be stored about each node, in order to be able to perform layout and mapping so that it may state to this specification. one item -- the location from the node within distance, i.e., a hyperbolic plane, to the child node -- a variation rate is shown. the item of another side -- the include angle in the hyperbolic plane between the escapes of the outgoing link from the close link and its parents to the parents of the node to the node -- it is a variation rate. The handle which can be used in order to access the data of these dyadic eyes or them can be contained in the data item of the link within directed graph DS.

[0107] The test of the decision box 380 applies suitable criteria, in order that even the next generation of a node may determine whether continue a layout or not. As mentioned above with regards to the processing box 374, criteria may be whether the include angle of the child node of arbitration is changed more than the difference in a small include angle like 0.00001. The layout should be continued when that is right.

[0108] With the processing box 382, or the walker routine 222 is extended, it pushes ID of each child node which is not a leaf in front of a queue. Since other child nodes do not have the child arranged, the marking round of them may be carried out with the processing box 382. If a node determines that the test of the decision box 370 already went round if the processing box 382 is completed, or when it determined that the test of the decision box 380 will not continue, before returning to the decision box 360, the node behind a queue is taken out with the processing box 384.

[0109] Drawing 8 is the processing boxes 312, 324, and 334 of drawing 6 , and shows whether the layout of the node-link structure changed how can be performed. In each case, as shown by the processing box 400, a layout answers and is started at the call led to a layout and mapping. However, the approach by which a layout is performed is based on the type of modification made by node-link structure so that it may be shown by the branch of the decision box 402.

[0110] When modification is modification of the orientation of the root node which answered the orientation event, it is the processing box 404 and the walker routine

222 can arrange a root node for the mathematics routine 226 to new orientation in front of a call, mapping, and paint. Except new orientation, a root node can be arranged, as mentioned above with regards to the processing box 352 of drawing 6 . Next, new orientation is used for mapping and changes expressional orientation.

[0111] As it may be generated in the response to a click event when modification is a stretch event, a drag event, and a bookmark event and edit is suspending, when it is non-animation edit, it is the processing box 410, and the walker routine 222 sets up the list of removal edits first, and next it is the processing box 412 and it arranges removal edit before mapping and paint. Next, with the processing box 414, the walker routine 222 sets up the list of additional edits, is the processing box 416 next and arranges additional edit in front of mapping and paint.

[0112] In this operation, the list of edits is set up based on the edit source list maintained by the various routines in the memory 214 containing the GURAFA routine 220 and the painter routine 224. Current operation is related also to the tree defined by the extended link. One pair of edit source lists called "colla tempestade PUSUTO links (CollapsedLinks)" and "extractives pan dead links (ExpandedLinks)" on these specifications can be set up with the processing box 330 of drawing 6 , including [therefore] the edit about the link chosen by the contraction demand and the extended demand, respectively. Other pairs called "RIMUBUDO links (RemovedLinks)" and "ADIDDO links (AddedLinks)" on these specifications include the edit about the link deleted and inserted, respectively. Two or more copies of an edit source list may exist in order to differ.

[0113] The list of removal edits set up with the processing box 410 is based on RIMUBUDO links, and, on the other hand, the list of additional edits set up with the processing box 414 is based on ADIDDO links. In case a list is set up by the processing box 410 or 414, the walker routine 222 accesses each edit in a suitable edit source list, and receives the suitable entry about the list of [under the **** setup for edit]. In each case, the edit in an edit source list is used, and the edit identifier which shows the node ID of the child node of the link of edit and the type of edit under activation comes to hand.

[0114] The parents of the child node are added after the list of nodes which does not already exist on a list and which was restricted and influenced. the parent node to which, as for parents, the child node was extended -- or the parent node of the child node -- current -- when nothing is extended, it is the parent node of the beginning of the child node. When related to the child node in which the edit from an edit source list does not have parents, the edit is placed after the list of nodes which must be

related to the root therefore by which the root node was influenced in that case.
[0115] Before accessing the next edit in the suitable edit source list used for the last of the iteration about the link from a source edit list repeatedly [of a degree], the child node of the link is also added to a child node list. Thus, it is repeatedly carried out about each edit of an edit source list until all edits are processed, in order to complete the list of the nodes and child nodes which were influenced.
[0116] Next, in the processing box 412 or 416, edit is arranged using a list, a sequence similar to the sequence of the processing boxes 354-382 of drawing 7 about each node in the list of influenced nodes is followed, the node of software is pushed in front of a queue rather than a root node from a list, and some change is made as follows with the processing box 372. As for the layout of the processing boxes 412 and 416, in addition to discernment of which child is a non-extending leaf, each child determines whether exist in a child node list. When that is right, a layout carries out the multiplication of the include angle about the child, and the radius under the weight. In the processing box 412, since weight is 0, with the processing box 374, an include angle and a radius are mostly arranged by the location before that 0, therefore the child disappears. With the processing box 416, since weight is 1, the child is arranged by the new location in whenever [full-size], and a total radius in the processing box 374.
[0117] Actuation of the processing boxes 410-416 may also be carried out within an animation sequence, in that case, removal edit may be processed in the part of the beginning of an animation sequence, and additional edit may be processed in the consecutiveness part of a sequence. On the other hand, as mentioned above with regards to drawing 6 , when non-animation edit mainly arises from the node generation in paint, the edit may be only additional edit and the edit under all present holds may be processed at each step of an animation sequence.
[0118] As the demand which animates modification according to individual of a large number which answered the insert/delete event like the demand which contracts or extends an element, or were demanded by such event is answered and it may be generated, when modification is animation edit, the walker routine 222 is the processing box 420, and the number of the elements first removed and added based on a source edit list is received. On the other hand, the number added can be obtained by adding the number of the elements in extractives pan dead links and ADIDDO links by the ability obtaining the number removed by adding the number of the elements in colla tempestade PUSUTO links and RIMUBUDO links. Next, with the processing box 422, the walker routine 222 assigns an usable animation step between a removal step

and an additional step, and also sets up the list of removal edit and additional edits in it like the processing boxes 410 and 414 of drawing 8 a little. Although simple allocation of an animation step is a half removal step and a half additional step, when all steps can turn into an additional step when there is no element removed, and there is no element added, it is the reverse.

[0119] It sets to removal edit in the processing box 422, and the setup of an additional edit list, and unless the node contracted or extended exists, the walker routine 222 can be performed as mentioned above with regards to the processing boxes 410 and 414. In the case of the node contracted or extended, in addition to the parents, the node itself is pushed after the influenced node list, next the child of the node instead of the node itself is added to a child node list. In other words, unlike other actuation of influencing only one node, it can be considered that contraction or an escape influences the node of two generations. The walker routine 222 sets up the list of two pairs, the influenced node and a child node, and one pair is used for removal edit and it uses one more pair for additional edit.

[0120] next, the animation step which removes a node performs by the loop formation started with the decision box 430 -- having -- the processing box 432 -- weight -- receiving -- weight -- using -- removal edit -- arranging -- an animation frame -- mapping -- and paint is carried out. the animation step which similarly adds a node by the loop formation started with the decision box 440 next performs -- having -- the processing box 442 -- weight -- receiving -- weight -- using -- additional edit -- arranging -- an animation frame -- mapping -- and paint is carried out. Before adding a node, by removing a node, the situation that the same node appears in two places within a single frame is prevented. This technique can ensure that the last weight is 0 or 1, respectively by performing the one last step by weight 0, after removing a node, and performing another last step by weight 1, after adding a node.

[0121] By lengthening a current removal animation step number from the removal animation number of steps with the processing box 432, and then dividing the difference by the removal animation number of steps, weight can come to hand so that weight may be set to 0 from 1 among a series of removal animation steps. Similarly, with the processing box 442, by adding 1 to a current additional animation step number, and then dividing the sum total by the additional animation number of steps, weight can come to hand so that weight may be set to 1 from 0 among a series of additional animation steps.

[0122] The total of an animation step helps the effect of the consciousness on object permanence animation on display with an animation rate. The total of an animation

step determines indirectly the rate which must move with regards to the field of the element to which the rate from which the field of the element removed or added changes was determined, therefore other elements were removed or added so that he can understand from explanation of the approach of obtaining above-mentioned weight. The direction with much animation number of steps assigned [that sufficient animation rate is maintained and] appropriately between a premise, then a removal step and an additional step is suitable producing object permanence.

[0123] When the technique of drawing 8 was performed at a number with a suitable suitable rate of animation steps, it made well consciousness of 1 set of nodes contracted and extended like the fan which it can be closed a little or can be opened. By adjusting the radius and include angle which were assigned to the node, or the node which is pressed out to infinity or was inserted or the deleted node was drawn in the parents grows and comes out from the parents, different consciousness which is pulled from infinity can be obtained. When the child seems to be pressed out to infinity when only one child is deleted from one child's group, and all children are deleted as one group like contraction, all children are able to seem to be drawn in parents. When the child looks [pull / from infinity] the same when one child is added by one group, and all children are inserted as one group like an escape, all children are able to seem to grow and come out from parents. Furthermore, at the rate adjusted so that a grandchild might be stabilized, and might be seen and only a child might be moved and seen, while the child is won over to parents, a grandchild can be pressed out to infinity.

[0124] As mentioned above, it is possible to animate in parallel modification according to two or more individuals shown within an edit list. Furthermore, a client 212 can be equipped with an interface so that it can require that a change according to two or more individuals should be made without animation through an interface. In other words, an edit list can continue being formed until the command which requires completion of edit is received. A command can require that edit should be animation as mentioned above, or should be completed without animation like refresh actuation. An edit list is eliminable if edit is completed.

[0125] In order to make animation easy, additional escape/contraction information is maintainable. For example, an additional flag can enable quick exchange with the expression of front structure, and the expression of the changed structure by enabling rapid decision about which element should be scanned about each expression.

[0126] Drawing 9 and 10 show the instantiation-sequence of an expression of node-link structure which is displayed by the operation as the present software, and are substantially [as the sequence mentioned above with regards to 8 from drawing

4] the same. The expression currently illustrated shows the organization chart and resembles the expression of 21 a little from drawing 17 of U.S. Pat. No. 5,619,632.

[0127] Drawing 9 shows the sequence of the expression which may be produced from the demand which contracts a node feature, and, on the other hand, drawing 10 shows the sequence of the expression which may be produced from the demand which extends a node feature. Moreover, the sequence connected through drawing 10 from drawing 9 may be produced from the demand which extends the clone of the node feature extended before so that it may explain below.

[0128] The expressions 500, 502, and 504 of drawing 9 contain the node feature showing the individual in an organization chart. When each node feature is displayed with the maximum size, it contains the rectangular title block which has identifier or nickname of the graphic expression of a person's face, and its person. there may be "small +" or small "-" for each to require an escape or contraction of a node feature in the bottom right corner of the title block of the rectangle of the node which has a descendant. These small symbols can change like [in the 1st animation frame] at the suitable time, after being chosen. If a node feature is displayed smaller than the maximum size, it can have the size to which the graphic expression was reduced, and a title block can be short-****(ed). If a node feature approaches an expressional periphery, it can become a small point and, subsequently can disappear.

[0129] The node feature 510 exists in each of expressions 500, 502, and 504, it is extended and shown by expression 500 and the child is shown with it by the small point near the periphery. The sequence of drawing 9 may be produced from the demand which contracts the node feature 510. This demand is answered, the child and grandchild of the node expressed by the node feature 510 are removed from node-link structure, and the transition to the expression with absent child and grandchild from expression 500 is animated. Expressions 502 and 504 are two middle expressions which may be displayed by such animation sequence.

[0130] In expression 502, the child of the node feature 510 just began to be lengthened to the node feature 510. However, since the child has spread from the child at the rate which offsets the rate lengthened toward the node feature 510, the grandchild of the node feature 510 stops at a predetermined location.

[0131] With expression 504, the child is further lengthened by near toward the node feature 510, and, on the other hand, a grandchild is continuing stopping at a predetermined location. Furthermore, since the child separated from the expressional periphery, he has sufficient field which can display the title block of each child's rectangle. The title block has produced the consciousness of the card or sheet which

was able to be opened to overlap and 1 set of flabellate forms based on the order by which paint was carried out.

[0132] Since all children converge on the node feature 510, the pass which the child of the node feature 510 follows is not parallel. If a child approaches the node feature 510, the fields of the node feature 510 will decrease in number, and the size of a graphic expression a person's face will be reduced temporarily.

[0133] The last expression with which the child or grandchild of the node feature 510 is not displayed at all follows expression 504 after some middle expressions the back potentially. The node feature 510 can understand this situation from the expression 540 of drawing 10 which seems to be visible within the last expression.

[0134] At drawing 10 , the node feature 550 of expressions 540, 542, and 544 which is extended in the expression 544 of drawing 10 and shown although it contracts by the expression 504 of drawing 9 , respectively and is shown is included, and the child is shown by the small point near the periphery. The sequence of drawing 10 may be produced from the demand which extends the node feature 550, when the node feature 550 seems to be visible with expression 504. This demand is answered, the child and grandchild of the node expressed by the node feature 550 are added to node-link structure, and the transition to the expression with which a child and a grandchild exist from expression 504 is animated. Expressions 540 and 542 are two middle expressions which may be displayed by such animation sequence, and expression 544 is the last expression.

[0135] In expression 540, the child of the node feature 550 just began to spread from the node feature 550. Since only a certain distance has still separated the child from the expressional periphery, he has sufficient field which can display the title block of each child's rectangle. Like the expression 504 of drawing 9 , the title block has produced the consciousness of the card or sheet which was able to be opened to overlap and 1 set of flabellate forms. However, since a child is lengthened toward a child from infinity at the rate which offsets the rate which spreads from the node feature 550, the grandchild of the node feature 550 is already in a predetermined location.

[0136] With expression 542, the child separated from the node feature 550 further, and has spread, and, on the other hand, a grandchild is continuing stopping at a predetermined location. Furthermore, since the child approached by the expressional periphery, he does not have sufficient field for displaying a title block, but has become a small point.

[0137] Finally by the expression 544, the child of the node feature 550 has arrived at

the location near an expressional periphery. A grandchild is not seen according to it being close to a periphery.

[0138] Since all children diffuse the pass which the child of the node feature 550 follows from the node feature 550, it is not parallel. Since the child is close to the node feature 550 when a child appears first with expression 540, the fields of the node feature 550 decrease in number, consequently the size of a graphic expression a person's face is reduced temporarily. The field of the node feature 550 increases as a child distributes from the node feature 550.

[0139] As mentioned above, the single sequence to which the sequence of drawing 9 and drawing 10 is connected, the node feature 510 is contracted first, and then the node feature 550 is extended may be generated. Although the node features 510 and 550 express the same node in a graph, since the node has two close links, the clone of the node is generated, and this may be produced [both], when generating twice within the tree expressed with drawing 9 and 10. If a user demands the escape of the node feature showing one clone, the response can include contraction of the node feature showing the clone of another side extended before. In fact, the branch is moved to a node 550 from a node 510. "+" in two node features and the symbol of "-" are changed appropriately.

[0140] Although operation similar to above-mentioned operation was well performed on the processor compatible [PC] with IBM, operation may be performed with other equipments which have the suitable processor of arbitration.

[0141] Although operation similar to above-mentioned operation was well performed using C++ language in the 32-bit Windows (Windows) environment, it may use other programming language and environments including a non-object-oriented environment, and may use other platforms, such as LISP (Lisp), a UNIX (Unix) environment, ANSI C, and a pascal (Pascal).

[0142] although operation similar to above-mentioned operation was well performed using the node-link data displayed in an XML conformity format and the existing experimental format, this invention is static or dynamic -- one of the inside of memory or networks are minded -- you may perform using the node-link data of accessible arbitration suitable type by the suitable approach of arbitration like.

[0143] Although it carried [use] out repeatedly [each / which operation similar to above-mentioned operation answers a navigation signal, and prepares and displays a series of one expression of a graph expressed or animated], this invention may be carried [use] out repeatedly [of the type of the others called by other signals or calls of a type].

[0144] Operation similar to above-mentioned operation was well performed using the navigation signal related to the expression of a single string expressed or animated with which it was received from the keyboard and the mouse, and node-link structure was displayed. However, this invention may be carried out, without using, using a navigation signal. For example, sorting from which the child of a certain node differs may be answered, or application of a different filter to the element of a certain structure may be answered, and you may make it move about an element. Moreover, this invention may be carried out using other navigation signals including the signal produced from selection of an item like the menu entry which requires an escape of the bottom of the shown enquiry of an escape and contraction signal of arbitration suitable type, or the exterior, node, or link, or selection of an item like the menu entry which requires an escape of the bottom of a current focus. A navigation signal may be related to space instead like the space generated by video game or the virtual reality environment which is not actually, or operating spaces other than a display, and a navigation signal may be instead generated by the suitable user input equipment containing the equipment of other classes for receiving a linguistic input like the pointing device of other classes and an alphabetic character or voice, and a gesture, or other user inputs of a format of arbitration. This invention may be carried out using the suitable animation technique of arbitration.

[0145] A node is expressed by features, such as an image of the shape of a circle, a rectangle, and an icon, and, as for above-mentioned operation, displays the expression of the node-link structure expressed by the line by which a link connects a node feature. However, this invention may be carried out about some modes of this invention using other suitable approaches of the arbitration expressing node-link structure including the operation in the expression including the hierarchy list of items which have the child of the indented item downward at least.

[0146] Although above-mentioned operation maintains the object permanence about the feature which lets the sequence of a step pass and expresses a share node and a link by displaying a similar feature at the interval of time amount and space small enough, using other techniques for maintaining object permanence, this invention has the object permanence about the feature to which only a share node expresses only a share link, and may be carried out. For example, in order to help offer of the object permanence of a migration element, blurring between locations, the sequence of the profile between locations, or other queues like other indicators in which migration between locations is shown may be used.

[0147] The field for the feature which expresses with above-mentioned operation the

element deleted or inserted decreases in number or increases, respectively, while the fields of a neighboring element increase or decrease in number. However, or this invention enters from a periphery, it may be carried out using other techniques of changing a feature showing the element deleted and inserted like other conventional animation techniques of arbitration including an usable technique, by PowerPoint (PowerPoint) of modification of the color which shows the migration feature, deletion, and insertion which appear in a periphery, fade-in, fade-out, or Microsoft Corp. (Microsoft). Furthermore, this invention may be carried out using the technique of increasing or decreasing fields other than the field of a neighboring element. Furthermore, it may be generated immediately and deletion may produce insertion without animation immediately similarly in the end of the animation with which migration of a neighboring element is compensated at the beginning of the animation with which migration of the neighboring element which may be animated is compensated without animation.

[0148] In above-mentioned operation, 1 set of elements are deleted, then 1 set of elements are inserted, and migration of an element deletes an element from a front location first, and is completed by next inserting an element in a new location. However, this invention may be carried out using the technique which deletes and inserts an element in parallel, and this invention may be further carried out using the technique which moves by the approach except deleting an element and inserting in a degree.

[0149] Although above-mentioned operation obtains layout data by a certain specific approach, this invention may be carried out by obtaining layout data by other approaches according [or] to carrying out, without using, using such a layout, or arranging the whole node-link structure according to an individual about each expression so that according to arranging node-link structure by other approaches.

[0150] Although paint is carried out in above-mentioned operation by the specific approach which node-link structure is mapped by the unit disk and is in a degree This invention may be carried out without using, using mapping. Or include what node-link structure is mapped to other suitable rendering space of arbitration including three-dimension rendering space and an operating space, and is displayed on other suitable operating spaces of arbitration. You may carry out so that node-link structure may be mapped and displayed by other suitable approaches of arbitration.

[0151] Above-mentioned operation is suitable for the display of an expression of a tree. This invention may be used for displaying the expression of node-link structure of other types like a general graph.

[0152] Although above-mentioned operation uses the node-link data containing the extended flag of the link for defining the tree in a graph using the memory management performed by a certain specific approach, this invention may be carried out using the node-link structure which is defined by other suitable approaches of arbitration and loaded to memory by the suitable approach of arbitration.
[0153] although above-mentioned operation can treat the directed graph containing a circulation directed graph, this invention changes the link of other types into the suitable combination of a directed graph -- or another appearance -- if -- you may carry out for the graphs of other types by supplying the protocol for mapping the structure of a graph to a tree. For example, the **** link between two nodes may be changed into one pair of owner ** links between the same nodes, or may assign a direction based on suitable criteria. Although there is an inclination confused visually since each set of an owner ** link generally circulates through the expression from which all **** links are changed into one pair of owner ** links as a result, this derangement may be conquered by displaying circulation by the option.
[0154] Processing is performed by above-mentioned operation with the ranking which may be changed in many cases. For example, in drawing 6 , a depth-first round may be performed rather than a width-of-face priority round.
[0155] In the same above-mentioned operation, although the part of some software is distinguished by GURAFA, a walker, a painter and a mathematics routine, and the list like a client, this invention may be carried out by the software constituted from a suitable approach of arbitration by other combination lists of hardware and software.
[0156] This invention was applied to offer of the interactive browser of node-link structure. This invention may be applied to various contexts by which node-link structure is visualized. Especially this invention may be applied to visualization of web related structure like the structure formed of 1 set of web pages or other web objects which were stored in the cache.
[0157] More generally this invention may be applied to offer of the browser for an organization chart, a file system hierarchy, a hypertext hierarchy, World-Wide-Web connectability structure, parts decomposition, SGML structure, or other big node-link structures of arbitration. This browser may be used for edit of the contents of structure or structure.
[0158] Although this invention has been described with regards to the example as software, this invention may be carried out using exclusive hardware.

[Translation done.]

TECHNICAL FIELD

[Field of the Invention] This invention relates to the display of the node-link structure which modification produces.

[Translation done.]

* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] Journal OBU visual run GEJIZU- and - computing magazine (Journal of Visual Languages and Computing), 1996, the 7th volume, and 33 pages -55 pages, A run ping (Lamping, J.) and RAO (it Rao(s)) "hyperbolic browser of R. : a big hierarchy The focal + context technique ("The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies") for visualizing" The

technique for displaying a hierarchy whose number of components increases exponentially is indicated the size of a component decreasing and increasing a radius as a component moves outside. A display is transformed smoothly and a node is changed with a core or a focus. The transition which animated between different views maintains the permanence of an object, and a user helps to adapt himself to change. For quick regeneration, a fringe can be drawn fewer, a line can be drawn instead of an arc, and a text can be omitted during an animation display. U.S. Pat. No. 5,619,632 of run pings (Lamping et al.) is indicating the browser using the similar technique for displaying node-link structure, and has mentioned that a browser may be used for edit of structure.

[0003] Systems - and - KOMPYUTAZU Inn Japan (Systems and Computers in Japan) 1993, the 24th volume, No. 8, 35-page -46-page MATSUURA (Matsuura, T.), Taniguchi (Taniguchi, K.), MASUDA (Masuda, S.) and Nakamura (Nakamura, T.) "the graph editor for the big tree which has browsing and zoom capacity () ["A Graph Editor for Large Trees] with Browsing and Zooming Capabilities"" is indicating the library program which makes possible the direct control of the big tree on a display. DS will be updated if a node or a subtree is added or deleted. That is, the layout about each node is calculated based on a graph layout algorithm, and each node and an edge are redrawn. If a node is added or deleted continuously, after all modification is made, only one renewal of a display can be performed. A view port depends updating on the mode which is renewal of automatic, the renewal of a part, or renewal of on demand one and to obtain including a global view and canvas. Although the renewal of a part resembles renewal of on demand one, a middle display node is displayed temporarily.

[0004] This invention deals with the problem in the display of the node-link structure of changing.

[0005] Node-link structure may change in some situations. data available to given time amount when structure is large only define a part of structure -- ********** . Although the definition with that the data which define perfect structure have not arrived or very large structure, and perfect structure is held, since memory is not large enough, this may be produced. In such a case, structure partial in memory is held, and it is changed, when additional data arrives or is taken out.

[0006] Another situation that node-link structure changes is the case where structure is the partial version of structure which makes a foundation. For example, a user can require that the specific node or specific link of a type should be chosen, in order for the filtered version to come to hand. It is answered, the node-link structure of omitting the node and link which are not chosen by filter criteria is generated, and a

large tooth space is given by a target node and a target link. Or structure may be a tree showing the directed graph (DG) structure of making a foundation, and two or more close links to the node in DG which has a child further exist in directed graph structure. Although the DG node is expressed with one approach by two or more parent nodes which have one of the close links of each in a tree, the child of the DG node is represented only with one of the parent nodes in a tree. A user's input may be answered, the parent node with which a child is represented may change, and a tree structure changes according to it (however, DG does not change).

[0007] Moreover, the foundation structure itself may change. For example, foundation structure may be answered and changed into processes other than the browser which is looking at the foundation structure. Or foundation structure may be answered and changed into a demand of the user through a browser interface.

[0008] The node-link structure of changing by these approaches of arbitration is called "dynamic node-link structure" on these specifications. therefore, this vocabulary changes not only in the basic node-link structure of changing or is static or dynamic, in itself -- the partial version obtained by the approach of being different from the basic node-link structure as which any are sufficient is also included.

[0009] The display technique and other Prior arts which are stated by the run ping and RAO may be led to the problem accompanying a structural change on display. A layout typically new about a part of structure which changed at least must be made, and then regeneration of it must be carried out. therefore, the expression of the structure which changed when change arose in structure on display using such a technique -- often -- abrupt -- or discontinuous transition exists. In some cases, a layout and regeneration are very late, and it cannot have a dialog on the structure and the effectiveness target from which the user changed. Furthermore, a layout and regeneration do not offer the approach of animation the transition to the old new structure which will raise an understanding of a user's transition from structure. Therefore, many of conventional display techniques only generate a satisfactory result to static node-link structure, and it is not suitable for dynamic node-link structure.

.....

[Translation done.]

* NOTICES *

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, some of conventional display techniques can animate the transition which extends or contracts a hierarchy list item. However, it is difficult to limit these techniques to an escape or modification of any one contraction, therefore to use widely to all transition.

[Translation done.]

* NOTICES *
JPO and NCIPI are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] This invention eases the problem in the display of dynamic node-link structure by offering the technique which makes it easy that a user understands transition of the larger range. Each technique displays an expressional sequence, the first expression expresses 1st node-link structure, and the last expression expresses the 2nd node-link structure which is change-bar JON of 1st node-link structure. The element shared according to the structure of the beginning and the last contains the migration element with which it is expressed by the feature

(feature) which has the location in which the first expression differs from the last expression. This technique displays at least one middle expression between the first expression and the last expression. Each expression in this sequence contains the feature showing the subset of the share element containing a migration element. The feature showing a migration element has object permanence through the sequence of a step.

[0012] In one mode of this invention, 2nd node-link structure is change-bar JON accompanied by at least one insertion in 1st node-link structure, and at least one deletion. For example, an element is deleted from one location and may be inserted in another location. Therefore, the technique called "deletion and insertion" on these specifications in which this mode was followed may be used in order to move an element, or in order to make [rather than] a complicated change including two or more deletion and insertion.

[0013] New deletion and an insertion technique display the sequence of an image to which the field of the element under deletion or insertion carries out change of reduction or one of increases in each of a sequential image, respectively. Compensation field change arises to a neighboring element at coincidence. When field change is chosen appropriately, the sequence of an image approaches the element which a neighboring element is deleting in deletion, and produces the impression to which the element under insertion pushes aside a neighboring element by its side in insertion.

[0014] In each case, the sequence of an image can also offer a suitable motion of the element under deletion or insertion. For example, in the deletion of an element which has a wedge-shaped field, before the element under deletion can come out from the edge which the wedge opened, or can be shrunken along with the wedge or its wedge of the begins to shrink, it can disappear.

[0015] Another mode of this invention is based on discovery of the technique which animates transition on a two-dimensional display. According to this technique, two migration elements follow the pass which is not an parallel straight line. For example, an element may move along with curvilinear pass and may move along with the independent pass which is not parallel.

[0016] Still more nearly another mode of this invention is based on discovery of the technique which animates modification according to individual of node-link structure in parallel. According to this technique, the signal which requires modification according to at least two individuals is received, and the modification data in which modification demanded based on this signal is shown come to hand. Next, the 2nd DS

which defines change-bar JON of node-link structure comes to hand using the 1st DS which defines modification data and node-link structure, and an expression can be displayed according to it.

[0017] Since this new technique makes it easy that a user understands the transition within dynamic node-link structure, it is useful. Furthermore, this new technique does not carry out count which time amount required for the layout of perfect node-link structure and activation of a display requires, but can use for and carry out increment layout and regeneration instead. Consequently, a user can have a dialog with dynamic node-link structure more effectively.

[0018]

[Embodiment of the Invention] The vocabulary which the following notional designs are helpful when you will understand the large range of this invention, if it reads with the notional design stated by U.S. Pat. No. 5,590,250 used for this specification as reference and No. 5,619,632, and is defined below has the semantics shown through this application containing a claim.

[0019] "Node-link structure" is the structure where each link is related to two or more nodes including an item distinguishable to a node and a link. A "graph" is the node-link structure where each link is related to two nodes. A "directed graph" is a graph whose node of another side the direction of [between the nodes related to each link] is shown, one node is the source or a "FUROMU node" of a link, and is the destination or a "two node" of a link. A "non-ring type directed graph" is a directed graph which does not offer the pass which returns from the node of arbitration to itself, when a link is followed in those shown directions. A "tree" is a non-ring type directed graph which has only one root node which offers only one pass which starts in a root node and is led to a non-root node about the non-root node of the arbitration in a tree when a link is followed in the shown direction.

[0020] The "element" of node-link structure is the node and link of node-link structure.

[0021] "A node-link expression" is an expression of node-link structure. For example, a link feature can express a link and a node feature can express a node.

[0022] When a certain feature showing an element has the spatial range of each dimension of the display displayed, the feature "has a field." For example, on a two-dimensional display, when a feature is prolonged into the dimension of both displays, the feature has a field.

[0023] The "neighboring feature" of the feature showing a certain element in a node-link expression is close with a certain feature including the feature showing

other elements from the median of the distance between other features to which a neighboring feature expresses the element of a certain feature and others.
[0024] When all a certain features showing the element in the expression in a sequence seem to be the same feature, the feature showing the element of node-link structure "has object permanence" through the sequence of an expression of node-link structure. For example, the feature in two continuous expressions of the arbitration in a sequence may be similar enough, and may be close to time amount and a space target enough so that they may be visible to the same feature. the feature which has the permanence of an object -- "pass -- following -- " -- it seems to move.
[0025] An "animation loop formation" is each repetitive actuation which displays an image repeatedly, and looks like continuation of the feature in each precedence image to which a degree is followed [repeatedly] so that according to object permanence in the feature in each image. An "animation cycle" is once [of an animation loop formation] repeatedly.
[0026] The element "is extended" when a certain element has at least one child in a node-link expression. ["which has "descendant within a node-link expression, or] When a certain element does not have a child at all in an expression, the element "does not have a descendant." The element "is contracted" when it does not have a descendant at all in an expression, in spite of basing a certain element on the node-link structure of making a foundation.
[0027] On these specifications, the vocabulary a "navigation signal" means the signal which shows that the user has an interest higher than other parts in the part with node-link structure. For example, an "extended signal" shows the demand which displays the graphic representation to which the expression of an element with a graph is extended, and, on the other hand, a "contraction signal" shows the demand which displays the graphic representation by which the expression of an element with a graph is contracted. It is possible by the point and click actuation which require that it should depend especially or the feature for which it chooses a bookmark etc. including the demand whose examples of other display a part of node-link structure in a specific location, and to which it pointed should be moved to a focal core.
[0028] Like insertion of one or more elements or deletion, like the migration which can be carried out with the combination of insertion and deletion, or copy actuation, when a certain signal requires modification of one or more elements in structure, as for the signal, "modification is required" within node-link structure. An escape and a contraction signal are the examples of the signal which requires modification.

[0029] When two change may be respectively made independently of another side, two modification "is individual."
[0030] In drawing 1 , a display 10 begins with the 1st expression 20 of 1st node-link structure, and shows the sequence of the node-link expression ended with the last expression 50 of the 2nd node-link structure which is change-bar JON which performed both deletion which may be produced from the signal which requires modification according to two individuals of 1st node-link structure, and insertion. A display 10 displays the middle expression 40 among expressions 20 and 50.
[0031] 1st node-link structure has the 1st level node expressed by the feature 22, and the 1st level node has two children who are the 2nd level nodes expressed by features 24 and 26. The child to whom it is expressed by the feature 24 is at the top of a branch 28, and a branch 28 contains all descendants including three children who are the 3rd level nodes. Similarly the child to whom it is expressed by the feature 26 is at the top of a branch 30, and a branch 30 contains all descendants including two children who are the 3rd level nodes.
[0032] the 2nd node-link structure shown by the last expression 50 -- many elements -- 1st node-link structure -- sharing -- **** -- the subset -- expressions 20, 40, and 50 -- it is expressed by the feature of all. A share element has the location by which each was stabilized through expressions 20, 40, and 50 including the node expressed by features 22, 24, and 26 and the link to which they are connected. The node to which a share element is similarly expressed by the features 32 and 34 in a branch 28, The node expressed by the features 36 and 38 in a branch 30, Each feature showing one of these nodes and links including the feature showing the link to nodes 32, 34, 36, and 38 It has a different location from the inside of the 1st expression 20 within the last expression 50, therefore these nodes and links are called a "migration element" on these specifications.
[0033] In the example shown in drawing 1 , change of the location of the feature showing a migration element is produced for reasons of having been changed at two points, in order that 1st node-link structure may acquire 2nd node-link structure. In order to delete one of the 3rd level nodes expressed within the branch 28 in order to obtain a branch 52 and to obtain a branch 54, the additional 3rd level node is added to the branch 30. It is not concerned with change of the location produced as a result in the modification list between 1st and 2nd node-link structures, but the feature showing a migration element has object permanence through an expressional sequence. Nodes 36 and 38 seem to follow and leave curvilinear pass, and, on the other hand, nodes 32 and 34 seem to follow the curvilinear pass which goes mutually

so that it may be shown by the arrow head in expression 40 and 50. Therefore, these nodes seem to follow the pass which is not an parallel straight line.

[0034] Although the object permanence which lets a sequence pass can be produced by various approaches, the technique shown in drawing 1 shows the example which produces object permanence by displaying a similar feature at the small interval of space and time amount. While being displayed continuously in the same location quickly, it is substantially the same, and each of features 22, 24, and 26 and the feature which expresses the link between them to a list can be continued, and secures object permanence. While each of the feature showing the link between them is also quickly displayed on features 32, 34, 36, and 38 and a list continuously in the location displaced slightly, it is possible for it to be substantially the same and to continue. The variation rate of the location of features 32 and 34 brings them close mutually, and compensates them with deletion of the 3rd level node which is the child of a feature 24. That is, features 32 and 34 are more nearly mutually [than the inside of the branch 42 by which deletion was performed to the inside of the branch 28 before deletion, and the beginning] close within the branch 52 from which the node was deleted so that it may illustrate. in order that the variation rate of features 36 and 38 may compensate insertion of the 3rd level node which is the child of a feature 26 -- features 36 and 38 -- alienation -- it moves. That is, features 36 and 38 are further estranged rather than the inside of a branch 30 within a branch 44, and the 3rd level node is inserted within the branch 54.

[0035] In drawing 2 , processing of the processing box 100 is started by displaying the 1st expression. The 1st expression contains the feature which expresses 1st node-link structure and expresses the subset of a 2nd node-link structure and shared element. A subset contains a migration element like the node expressed with one of the features 32, 34, 36, and 38 of drawing 1 .

[0036] Next, processing of the processing box 102 displays the middle expression containing the feature which similarly expresses the subset of a share element. As shown by the dotted line, processing in the processing box 102 may be performed once or more, in order to display a series of two or more middle expressions. Within each middle expression, each feature showing a migration element has object permanence with the feature showing the migration element in the precedence expression which is the 1st expression or a middle expression to precede, and is obtained.

[0037] Finally, processing of the processing box 104 displays the last expression showing the 2nd node-link structure which is change-bar JON of 1st node-link

structure. Like the processing box 102, it has object permanence with the feature showing the migration element in the middle expression which the feature showing a migration element is not concerned with change of the location from the 1st expression to that last table current, but precedes the last expression including the feature showing the subset of a share element.

[0038] The equipment 150 of drawing 3 contains the processor 152 connected in order to supply the data which define an image as a display 156, while receiving the data in which a user signal is shown from the user input circuit 154. The processor 152 is connected also in order for the 2nd structure to access the node-link data 158 which define the 1st and 2nd node-link structure which is change-bar JON of the 1st structure. The processor 152 is connected also in order to receive the instruction data 160 in which an instruction is shown through memory 164, storage access equipment 166, or the instruction input circuit 162 that can supply the instruction received from connection with a network 168.

[0039] In activation of the instruction shown by depending instruction data 160, a processor 152 displays the 1st expression of the 1st node-link structure containing the feature which expresses the subset of a 2nd node-link structure and shared element to a display 156. The subset of a share element contains a migration element as mentioned above. Next, similarly a processor 152 displays on a display 156 at least one middle expression containing the feature to which each expresses the subset of a share element. That is, the feature showing a migration element has object permanence with the feature showing the same share element in a precedence expression. Finally, the feature which a processor 152 makes display the last expression of the 2nd node-link structure which contains in a display 156 the feature which similarly expresses the subset of a share element, and expresses a migration element has object permanence with the feature showing the migration element in the middle expression to precede.

[0040] As mentioned above, drawing 3 shows the three possible sources 164 which can receive the data which the instruction input circuit 162 shows an instruction, i.e., memory, storage access equipment 166, and a network 168.

[0041] the conventional memory of the arbitration in equipment 150 in which memory 164 contains random access memory (RAM) or read-only memory (ROM) -- or you may be the circumference or the remote memory apparatus of a class of arbitration.

[0042] Storage access equipment 166 may be the drive for accessing the storage 170 which are other suitable media of the arbitration for storing or more 1 sets of one tapes, a diskette or a magnetic medium like a floppy disk, an optical medium like or

more 1 sets of one CD-ROMs, or data, and is obtained, other suitable equipments, or a circuit. A storage 170 may be some equipments 150, a part of server, other circumferences, a remote memory apparatus, or a software product. In each in these cases, a storage 170 is one product which can be used into equipment 150. It is able to arrange a data unit, and for storage access equipment 166 to access a data unit, and to enable it to supply them by the sequence through the instruction input circuit 162 at a processor 152 on a storage 170. If a data unit is supplied by the sequence, it will form the instruction data 160 to which it is indicated that it is illustrated by instruction.

[0043] A network 168 can supply the instruction data 160 received from equipment 180. The processor 182 in equipment 180 crosses a network 168, and can establish connection with a processor 152 through the network connection circuit 184 and the instruction input circuit 162. Which processor may start connection and connection may be established with the suitable protocol of arbitration. Next, since a processor 182 accesses the instruction data stored in memory 186, crosses a network 168 and can transmit instruction data to a processor 152, a processor 152 can receive the instruction data 160 from a network 168. Next, it is stored in memory 164 or somewhere by the processor 152, and the instruction data 160 can be performed.

[0044] Drawing 4 shows how modification according to individual can be processed in parallel.

[0045] Processing of the processing box 190 receives the signal which requires modification according to at least two individuals in 1st node-link structure. For example, a signal may also include two different signals which requires modification according to individual.

[0046] Based on this signal, processing of the processing box 192 receives the modification data in which modification demanded by the signal is shown. Modification data may be stored by the suitable DS according to use.

[0047] Next, processing of the processing box 194 receives the 2nd DS which defines 2nd node-link structure using the 1st DS which defines modification data and 1st node-link structure. 2nd node-link structure is the version by which modification required of 1st node-link structure was performed.

[0048] Finally, processing of the processing box 196 displays the expression of 1st and 2nd node-link structure with animation. This processing can include the processing mentioned above with regards to drawing 2 .

[0049] The general feature mentioned above may be carried out on various equipments by many approaches for displaying a node-link expression. The example

described below runs the 32-bit version of Microsoft Windows (Microsoft Windows), and is carried out on the system based on PC which performs the code compiled from the C++ language source code.

[0050] In drawing 5 , a system 200 contains the PC processor 202 connected to the mouse 208 and keyboard 206 for supplying a signal from the display 204 for displaying an image, and a user. The PC processor 202 is connected also so that memory 210 and a client 212 can be accessed. Memory 210 can contain program memory 214 and data memory 216 as illustrated. the combination of the routine and data with which the client 212 was stored in memory 210 -- or it is the source of the information about a directed graph which may be independent of memory 210 so that it may be shown. For example, a processor 202 may communicate with a client 212 through a network.

[0051] The routine stored in program memory 214 can carry out grouping to some functions. The GURAFA routine 220 generates and changes the DS showing the directed graph defined using the information from a client 212. The walker routine 222 answers a navigation signal and other user signals from a keyboard 206 and a mouse 208 by information coming to hand from directed graph DS. The painter routine 224 provides a display 204 with a signal, and displays the expression of directed graph DS on it. The mathematics routine 226 can be called in order to obtain the location of the element of the directed graph in layout space.

[0052] Next, data memory 216 holds the DS accessed by the processor 202 during activation of the routine in program memory 214. As mentioned above, directed graph DS 230 may be accessed by the walker routine 222 and the painter routine 224 while it may be generated and changed by the GURAFA routine 220.

[0053] It is linked in directed graph DS 230, and it can obtain or the node location data 232 which may be contained can include the location of the node in **** space like a hyperbolic plane, and rendering space like a two-dimensional unit disk (unit circle). The node location data 232 may be accessed by the routine in program memory 214.

[0054] The routine in program memory 214 can also access various coarse DS 234. DS 234 may also include the preliminary data structure for mapping to Link ID from one pair of nodes ID carried out as for example, a standard heap. That is, this preliminary data structure enables the retrieval and insertion of Link ID within fixed expected time.

[0055] Drawing 6 shows how an event can be answered when the system of drawing 5 displays the expression of a graph.

[0056] With the processing box 300, a client 212 is started by loading the initial set of

an element to memory, for example through the call of node generation while an initiation graph comes to hand. An extended flag defines the tree in the initial set of an element. A client 212 also performs the suitable call to the routine in memory 214, in order to perform all the things for which the version by which paint was carried out by arranging a tree to a hyperbolic plane, carrying out the paint of the tree which mapped the root node from the hyperbolic plane at the core of a disk to the unit disk, and was mapped in the tree, and carrying out the swapping of the double buffers is displayed on a display 204 with the processing box 300.

[0057] With the processing box 302, a client 212 receives the event related to a graph. An event may be produced from a navigation signal, an edit signal, or the signal of other types of a user. Or an event may be received from other sources of the interior of a system 200, or one of the exteriors. Anyway, an event can take the gestalt of the call from other instructions executed by the processor 202 from one of the routines in [out of a client 212] memory 214. The event by which a single string was received may be held in a queue so that the processing box 302 may include the ejection of the event from a queue.

[0058] A client 212 starts a suitable response by answering the event received with the processing box 302 by performing one or more calls to the routine in memory 214. Since a response is based on the type of an event so that it may be shown by the decision box 304, a branch is chosen based on an event.

[0059] An event may be a non-animation event like an orientation conversion event, a stretch event, or a drag event. An orientation event may be produced when a user shows new orientation to a root node. A stretch event may be produced when a new stretch factor is shown to the expression with which the user is displayed. A drag event may be produced, when a user chooses the location in an expression by mouse down (have pressed mouse button) click etc. and demands migration of the location with suitable gesture or other suitable signals.

[0060] With the processing box 310, a client 212 starts the response to a non-animation event by the information on arbitration required for the response to an event coming to hand. About an orientation event, the information which came to hand with the processing box 310 can include new orientation. About a stretch event, the information which came to hand with the processing box 310 can contain a new stretch factor.

[0061] About a drag event, complexity of acquisition of the information on the processing box 310 increases a little. A client 212 may receive the information about the motion which could receive the node identifier (node ID) of the nearest node from

the selected position, and was demanded. these informational items are indicated with regards to the column 71-72 and drawing 14 of U.S. Pat. No. 5,590,250 which are used for this specification as reference -- " -- the nearest node may come to hand by the almost same approach as it is explained by the function (function) retrieval."

[0062] A client 212 can be rounded off with the suitable call to the walker routine 222 and the painter routine 224 for a layout, mapping, and paint if a client 212 receives required information with the processing box 310. About an orientation event, a root node must be arranged for new orientation. About a stretch event or a drag event, a layout is unnecessary. However, about a stretch event, the call to the walker routine 222 must contain the new stretch factor for using for mapping. Similarly, about a drag event, the call to the walker routine 222 must include the location of a degree which met the node ID of the nearest node for using for mapping, and the pass of a motion.

[0063] With the processing box 312, first, the walker routine 222 may perform the required layout of arbitration in a hyperbolic plane, and may arrange the edit under hold of the arbitration of a tree. Next, with the processing box 314, the walker routine 222 may start an initiation node in a starting position, and may map a tree on a unit disk. For example, a drag event may be answered, an initiation node may be nearest node identified with the processing box 310, and a starting position may be a location of a degree which met the pass of a motion. The initiation node and starting position which were before used for mapping may be answered and used for orientation or a stretch event.

[0064] If a tree is mapped, in order to carry out the paint of the mapped tree to a display buffer with the processing box 316, the painter routine 224 can be called. The painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation among paint. Each edit can carry out marking by [which set up a flag] depending especially or storing other suitable data. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph will be supplied.

[0065] As mentioned above, current is carried out for these events as a non-animation event. Answering an orientation event, an expression circles in the surroundings of the node which is in new orientation typically at the focus of a viewing area. Similarly, a stretch event is answered, and an expression is extended or contracted around the node which is in radial typically at a focus. Answering a drag event, an expression moves at the rate determined by the input signal. However, a client 212 may supply the response animated in demanded modification by depending, changing into the sequence of a small event, being the processing box 310 and

emitting a series of equivalent calls to an event with one call smaller than each one to an orientation event, a stretch event, or a drag event.

[0066] Drawing 6 also shows the response to the event of two different types which always animates and are treated by the present operation. The 1st type is a bookmark or a click event, it is answered, the location of one node is moved into an animation sequence, and other elements move it according to a motion of the one node. Although one node has been stabilized in the animation sequence to which the 2nd type is an insert/delete event, it is answered, a certain element is contracted, and another element is extended, other elements move to compensate for contraction and an escape.

[0067] A bookmark or a click event may be produced, when the item in the set of a user of a menu or other bookmarks is chosen, or when the location in an expression is chosen by mouse down rise (mouse button is pressed and released) click. Answering this type of event, a client 212 receives the destination location in Node ID and a unit disk. In the case of the bookmark event, it is stored before, and it can search Node ID and a destination location from memory. in the case of the click event, the client 212 is indicated with regards to the column 71-72 of U.S. Pat. No. 5,590,250 used for this specification as reference -- " -- it may be the almost same approach as the function [node / nearest] (function) retrieval", and the node ID of the nearest node may come to hand from a selected position, and the destination may be a default location like the core of a unit disk.

[0068] With the processing box 320, a client 212 may call the walker routine 222 in Node ID and a destination location. The walker routine 222 can answer by executing the animation loop formation for displaying the sequence of the expression which a node moves to a destination location from a front location. With the processing box 320, the walker routine 222 is started by setting up a pair of sequence of the node/location where each includes the location in Node ID and a unit disk. A location can come to hand by repeating the total translation (migration way) from a location to a front destination location coming to hand, and then the n-th root of the total translation coming to hand, and constituting with the present translation as indicated in the boxes 470, 472, and 482 of drawing 12 of U.S. Pat. No. 5,619,632. A pair of number of a node/locations can be done greatly enough, in order to secure the smooth animation from a location to a front destination location, while the feature which expresses the element of structure during an animation display maintains object permanence. A location may come to hand by choosing a suitable number of points from a front location instead of the approach of the n-th root along with the arc

appropriately chosen in the hyperbolic plane in the destination location. An arc may be chosen so that between the straight line which may look unnatural, and the arcs which the node probably took by the approach of the n-th root and which may need too much animation number of steps for looking smoothly may be compromised. The number of points may be chosen so that satisfactory animation may be secured.

[0069] Next, the walker routine 222 repeats an animation loop formation about each set of the node/location in a sequence, as shown by the decision box 322. With the processing box 324, the walker routine 222 may arrange the edit under hold of the arbitration of a tree to a hyperbolic plane first, as mentioned above with regards to the processing box 312. Next, with the processing box 326, the node and location from the pair of a node/location of a degree may be started as an initiation node and a starting position on a unit disk, and the walker routine 222 may map a tree on it, as mentioned above with regards to the processing box 314.

[0070] If a tree is mapped, in order to carry out the paint of the tree mapped with the processing box 328 to a display buffer, the painter routine 224 can be called. Among paint, the painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation, as mentioned above in relation to the processing box 316. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph can be offered.

[0071] If marking of the new edit is carried out by the painter routine 224 with the processing box 328, new edit will be arranged with the processing box 324 during the next iteration. Consequently, the sequence by which the expression was animated shows dynamic node-link structure rather than shows static node-link structure like U.S. Pat. No. 5,629,632. However, since fundamentally adds the feature which expresses a new node along with an expressional periphery when an expression changes from a front location to a destination location, this edit works. Consequently, the consciousness of the object permanence about the feature showing other elements blocks, or the added feature does not reduce it.

[0072] An insert/delete event may be produced, when a user demands an escape or contraction of a node, or when modification of everything but some of graphs or trees is required. An insert/delete event may offer the device for automatic modification of the graph which it may be received [graph] also with the gestalt of a call, therefore does not make control by people's hand concurrent, or a tree. Furthermore, an insert/delete event may be produced from the signal which requires modification according to two or more individuals which has data in which modification which

suspended the call to the animation of all modification and was obtained is shown. One insert/delete event may also include both deletion and insertion.

[0073] This type of event is answered, and it is the processing box 330, and a client 212 can perform the suitable call to the routine in memory 214 first, in order to determine whether modification of the demanded graph or tree is permissible.

[0074] When modification of the demanded graph or tree can be permitted, according to it, a client 212 calls the routine in memory 214 if needed, and can change a graph or a tree. In the process into which it changes, marking of each element called "the element by which it was influenced" is carried out by a setup of a flag, or storing of other suitable data in these specifications which insert and delete or may be changed by modification. Since the field assigned to the parents may change when a certain node is chosen by the extended signal or the contraction signal, it is the node by which the parents were also influenced. Only the node inserted or deleted is influenced about most of other insert/delete events. A certain node can choose a client 212 as a stabilization node held also during the animation display of modification in the location in front of the node.

[0075] In many cases, the initiation node used for mapping can be chosen as a stabilization node in front, and the node can be held to a front starting position. In other cases, selection of a different stabilization node is sometimes desirable for a client 212. For example, the node currently extended may be chosen as a stabilization node held in a current location, therefore the node serves as a new starting position. Therefore, unless the stabilization node from which a client 212 differs is chosen, a front initiation node and a front starting position are usually maintained. However, when the front initiation node is deleted, other nodes must be chosen as a default stability node changed by the client 212.

[0076] When deletion is performed, the walker routine 222 can be called by the node ID of the nearest ancestor that remains even after deleting in the node under deletion, and the tree under mapping. This ancestor can find it by what (walk) is patrolled upward until it reaches the ancestor who is not deleted from the node under deletion by the present insert/delete event.

[0077] This call is answered and, as for the walker routine 222, the node under deletion can test whether it is a front initiation node. When that is right, the identified ancestor can be chosen as an initiation node in order to permute by the node under deletion. It is mapped by the location where the ancestor is displayed recently, and when it is an usable location, the location can be chosen as a starting position. When it is mapped by the unusable location since the location which is not displayed, or the

element besides the present are mapped or an ancestor was not mapped recently, a starting position can be set as the core of a unit disk.

[0078] Also in the processing box 330, a client 212 may call the walker routine 222 in ID and the location of a stabilization node. The node from which the walker routine 222 was deleted first is contracted in a front location, the node inserted in the degree is extended in a new location, and it can answer in the meantime by executing the animation loop formation which displays the expressional sequence that a stabilization node is held in a front location, all the time. Since a stabilization node is mapped by the location or the unusable location which is not displayed or it was not mapped recently, when it cannot hold in a front location, it is in the condition which has a front initiation node in a front starting position, and after the deleted node is contracted, a stabilization node can move to the location, consequently the sudden motion between contraction and an escape arises. The walker routine 222 is started by setting up the sequence of weight which governs the rate from which the field assigned to each influenced node during contraction and an escape changes. Weight is classified for the increment small enough, in order to maintain object permanence animation on display.

[0079] Next, the walker routine 222 repeats an animation loop formation about each weight in a sequence, as shown by the decision box 332. With the processing box 334, the walker routine 222 may arrange the edit under hold of the arbitration of the node and tree which were influenced first in the hyperbolic plane using the weight of iteration. Next, in the processing box 336, the walker routine 222 may start and map a tree in a stabilization node and a location on a unit disk, as mentioned above with regards to the processing box 314.

[0080] If a tree is mapped, in order to carry out the paint of the mapped tree to a display buffer with the processing box 338, the painter routine 224 can be called. Among paint, the painter routine 224 can carry out marking of the new edit produced within a tree as a result of node generation, as mentioned above with regards to the processing boxes 316 and 328. If paint is completed, in order to display a tree that paint was carried out, the swapping of a display buffer can be performed, and the expression of a graph can be offered.

[0081] Even if the edit under hold exists and it does not carry out, the expression of the dynamic node-link structure by deletion and/or insertion is generated repeatedly [of a single string of the animation loop formation started with the decision box 332]. Furthermore, the influenced element moves to a new location from the location before deletion and insertion. This technique was well realized, in order to produce the object permanence between these motions.

[0082] After the expression was supplied with the processing box 316, or after an animation sequence is completed by the decision box 322 or 332, another event is receivable with the processing box 302 so that it may be shown by the circle displayed as "A" of drawing 6 .

[0083] Drawing 7 shows how a layout can perform first with the processing box 300 of drawing 6 . Drawing 8 is the processing boxes 312, 324, and 334, and shows how the layout of the changed node-link structure can perform.

[0084] As shown by the processing box 350, the walker routine 222 starts the first layout by using in order for a root node ID to come to hand and to access the data related to the root node within directed graph DS 232. With the processing box 352, the walker routine 222 arranges a root node by performing the call to the mathematics routine 226 by include-angle width of face. This include angle is good at the suitable include angle of the arbitration which generates a desirable result. When include-angle 2pi and pi/2 were used well and they used 2pi, it was suitable for the main layout style, and when pi/2 was used, it was suitable for the right or a left layout style a top and the bottom. An interface may be offered, in order to obtain a desirable result and to correct this include angle.

[0085] By response, the mathematics routine 226 arranges a root node at the zero of a unit circle, i.e., a coordinate, (0 0) for upward orientation at an angle of the one half of a coordinate (0 1) and include-angle width of face. Next, the walker routine 222 pushes a root node ID in front of a queue with the processing box 354.

[0086] In the remainder of drawing 7 , as shown by the decision box 360, the walker routine 222 scans repeatedly 1 set of elements of the tree defined according to directed graph DS 232 until a queue becomes empty. It is started by repeatedly [each] Node ID coming to hand after a queue by using in order to access at the data related to the node from which it was discriminated within directed graph DS 232.

[0087] It is determined [each] whether, repeatedly, the node already patrolled the test in the decision box 370 by this scan. When not going round, the walker routine 222 carries out marking of the gone round round node with the processing box 372. So that two or more children's node ID comes to hand, the non-extending leaf of two or more children, i.e., a leaf node without the extended close link, may be identified, the number of the children who exist in a visible tree may come to hand so that it may explain below, and it may similarly explain below In order for the include angle about two or more children in a visible tree and the array of a radius to come to hand, the mathematics routine 226 is called by several Ns of the child of the node which exists in a visible tree.

[0088] About processing in the processing box 372, several Ns of the child in the visible tree which is one type of neighboring relational data are calculable with either of two kinds of approaches. That is, when a current scan is a part of sequence of a node addition step, N is equal to the number of the children before a scan, and a child's added number of sum totals. When a current scan is a part of sequence of a node removal step, N is only equal to the number of the children before a scan.
[0089] With the processing box 372, the array of an include angle and a radius can come to hand by various approaches. At one successful example, it is set as the value of 0.7 and, on the other hand, each radius is each include angle. (N*pi) (/18) It is set as the smaller one with pi. Therefore, the include angle of a node is influenced about N< 18 by the number of the children of the node which exists in a visible tree.
[0090] Next, with the processing box 374, the walker routine 222 calls the mathematics routine 226, in order to arrange a child.
[0091] In case the processing box 374 is carried out, two general principles of a layout are applied. The alienation between nodes and an include angle are determined as the 1st only based on the information about a neighboring element, i.e., the neighboring relational data, in a tree. The layout information received about each node by the 2nd is the approach which the location of a certain node and all its children can move by the small change within DS, and shows the relative position of the parents of node opposite **.
[0092] The general strategy which can follow is that start at a child's radius and include angle from the processing box 372, the approximation distance which a child needs comes to hand, the distance from parents comes to hand using approximation distance, and a more exact distance about a child comes to hand using the distance from parents, next a further more exact distance comes to hand using a more [in optional] exact distance etc.
[0093] When a child has the radius R and include angle theta from the processing box 372 according to a general strategy, the approximation distance D1 and D2 can be calculated as sinh (R) and tan (theta/4), respectively. When N is the number of the children under layout activation and the sum total is performed from i= 1 to N-1 The pair of each child who approached A larger distance of each sum total of D1 and D2, Namely, DT=sigma (when divided by max(D1(i)+D1 (i+1), D2(i)+D2 (i+1))+max(D1 (1), D2 (1))+max (D1 (N), D2 (N))) The total distance DT about all children can be acquired using D1 and D2. When the parents have the usable include angle omega, the distance DP from parents can be calculated as asinh (DT/omega). Next, a child can be stationed the include angle between the children proportional to division of a child in

accordance with the periphery of the circle of the radius DP centering on parents.
[0094] Next, a more exact distance about a child can be acquired as follows using DP. D1 -- '=sinh(DP) asin (sinh(R)/sinh (DP)), D2' -- even parents' more exact distance DP' can be obtained until it reaches the accuracy of desired level like an above-mentioned approach using =2sinh(DP) atan (tan (theta/4)/eDP) next D1', and D2'. At the time, each child's orientation is calculable as a gap of the include angle from parents' orientation.
[0095] Please note that an above-mentioned distance is expressed with actual measuring in a hyperbolic plane. The distance D in a hyperbolic plane corresponds to the vector which begins at the zero in a unit circle and progresses the distance of tanh (D/2).
[0096] As mentioned above with regards to the processing box 372, a general strategy is carried out in this way, and receives the layout information only based on the neighboring relation information about a sibling including the information about a node, its parents, and the child of the sibling in a visible tree. A general strategy receives the layout information which shows the distance from a child to the parents, and the include angle showing the difference in the orientation of a child and its parents.
[0097] The general strategy was carried out as software using the first distance acquired without carrying out the attempt which acquires a more exact distance by the approach mentioned above with regards to the general strategy, although two repetitive loop formations were carried out about all children. The 1st loop formation receives the separation between the adjoining children, and the "slice size" about each child, is saved temporarily and also receives the sum total of separation. Next, the distance to parents is acquired using this information. Next, the 2nd loop formation receives and saves each child's relative orientation and relative field.
[0098] In the example as this software, when a child has the radius R and include angle theta from the processing box 372, as mentioned above with regards to the general strategy, distance D1 and D2 is calculated. D1 and D2 about each child are added to D1 and D2 about the former child, and S1 and S2 are obtained. The sum total ST of separation is just increased to the child of the beginning and the last by a child's maximum of S1 and S2 except for **, and it is increased to the child's maximum of D1 and D2 by ST about the child of the beginning and the last.
[0099] When S1 of a child is larger than S2 of the child, while S1 is saved as separation of a child, D1 is first saved as a child's slice size, and about the child of the 2nd henceforth, it is adjusted so that the former child's slice size may serve as the minimum value of the slice size and S1 of a child in front of the child. On the contrary,

when S1 of a child is not larger than S2 of the child, while S2 is saved as separation of a child, D2 is first saved as size of a child's slice, and about the child of the 2nd henceforth, it is adjusted so that the former child's slice size may serve as the minimum value of the slice size and S2 of a child in front of the child. However, the last child's slice size is adjusted so that it may become the minimum value of the slice size and a child's maximum of D1 and D2 in front of the child, and the 1st repetitive loop formation completes it.

[0100] Next, the distance DP from parents is calculable using parents' include angle omega as tanh (asinh (ST / 2omega)/2) or the larger one of 0.5. DP is saved as some data relevant to a parent node.

[0101] About each child, the 2nd repetitive loop formation is started by calculating the include angle (S/ST) of 2omega, when S is the separation saved about the child. The include angle (S/ST) of 2omega is applied to the running total started by -2omega. A running total is saved with other data relevant to a child.

[0102] Next, the mathematics routine 226 can calculate the new include angle about a child by calling a function similar to the "interior angle (inside-angle)" function of the columns 67 and 68 of U.S. Pat. No. 5,590,250. On these specifications of these, the function called "an inside angle type (InsideAngle)" is started at an angle of the one half of the distance ("dist") and the wedge which were moved into the wedge. An inside angle type takes the smaller one of an initiation include angle and (pi-epsilon) in case epsilon may have a very small value like 0.0001 as an include angle which can be operated, and avoids the problem in count of an inverse tangent. An inside angle type receives the deformation which moves the point in the coordinate on a unit circle (dist, 0) to a zero. Next, an inside angle type applies this conversion to the complex-locus label of the point on the intersection of the periphery of a unit circle, and the radiation which is emitted from a zero and whose include angle with a horizontal position is an include angle which can be operated. An inside angle type returns the include angle from the horizontal position of the radiation which passes along the point changed from the zero as an include angle of a result.

[0103] In order to obtain a child's include angle, an inside angle type is called at the include angle calculated by carrying out the multiplication of distance DP and the slice size of the child from the 1st iteration by 2omega / ST. The include angle returned by the inside angle type is measured with pi/2, and a child's include angle is the smaller one of these two.

[0104] Before saving a child's new include angle, the mathematics routine 226 saves the include angle in front of a child. When the absolute value of the difference of an old

include angle and a new include angle exceeds the minimum value, the mathematics routine 226 also saves the data in which it is shown that a layout should continue so that it may state below.

[0105] the last -- the 2nd iteration loop formation -- "space usable (room-available)" of the columns 67 and 68 of U.S. Pat. No. 5,590,250 -- a child's field or side tooth space is obtained by calling a function similar to a function (function). On these specifications, this function called "available [room available (RoomAvailable)]" is started at an angle of [phi] the one half of the distance D and the wedge which were moved into the wedge. room available -- first -- a ratio (1-D2) -- by obtaining /2D, in order to acquire the initial distance S next, the distance to the edge of a wedge calculated by breaking a ratio by sinphi is returned. Next, room available returns distance (S2-1) (1 / 2-S). In order to obtain a child's field, room available is called at the same distance and the same include angle as having been used for the call of an above-mentioned inside angle type. The distance returned with room available is saved as a scale of a child's field.

[0106] Although the example as above-mentioned software can save additional data, the example as software is based on discovery that only the data of a dyadic eye need to be stored about each node, in order to be able to perform layout and mapping so that it may state to this specification. one item -- the location from the node within distance, i.e., a hyperbolic plane, to the child node -- a variation rate is shown. the item of another side -- the include angle in the hyperbolic plane between the escapes of the outgoing link from the close link and its parents to the parents of the node to the node -- it is a variation rate. The handle which can be used in order to access the data of these dyadic eyes or them can be contained in the data item of the link within directed graph DS.

[0107] The test of the decision box 380 applies suitable criteria, in order that even the next generation of a node may determine whether continue a layout or not. As mentioned above with regards to the processing box 374, criteria may be whether the include angle of the child node of arbitration is changed more than the difference in a small include angle like 0.00001. The layout should be continued when that is right.

[0108] With the processing box 382, or the walker routine 222 is extended, it pushes ID of each child node which is not a leaf in front of a queue. Since other child nodes do not have the child arranged, the marking round of them may be carried out with the processing box 382. If a node determines that the test of the decision box 370 already went round if the processing box 382 is completed, or when it determined that the test of the decision box 380 will not continue, before returning to the decision box 360,

the node behind a queue is taken out with the processing box 384.
[0109] Drawing 8 is the processing boxes 312, 324, and 334 of drawing 6 , and shows whether the layout of the node-link structure changed how can be performed. In each case, as shown by the processing box 400, a layout answers and is started at the call led to a layout and mapping. However, the approach by which a layout is performed is based on the type of modification made by node-link structure so that it may be shown by the branch of the decision box 402.
[0110] When modification is modification of the orientation of the root node which answered the orientation event, it is the processing box 404 and the walker routine 222 can arrange a root node for the mathematics routine 226 to new orientation in front of a call, mapping, and paint. Except new orientation, a root node can be arranged, as mentioned above with regards to the processing box 352 of drawing 6 . Next, new orientation is used for mapping and changes expressional orientation.
[0111] As it may be generated in the response to a click event when modification is a stretch event, a drag event, and a bookmark event and edit is suspending, when it is non-animation edit, it is the processing box 410, and the walker routine 222 sets up the list of removal edits first, and next it is the processing box 412 and it arranges removal edit before mapping and paint. Next, with the processing box 414, the walker routine 222 sets up the list of additional edits, is the processing box 416 next and arranges additional edit in front of mapping and paint.
[0112] In this operation, the list of edits is set up based on the edit source list maintained by the various routines in the memory 214 containing the GURAFA routine 220 and the painter routine 224. Current operation is related also to the tree defined by the extended link. One pair of edit source lists called "colla tempestade PUSUTO links (CollapsedLinks)" and "extractives pan dead links (ExpandedLinks)" on these specifications can be set up with the processing box 330 of drawing 6 , including [therefore] the edit about the link chosen by the contraction demand and the extended demand, respectively. Other pairs called "RIMUBUDO links (RemovedLinks)" and "ADIDDO links (AddedLinks)" on these specifications include the edit about the link deleted and inserted, respectively. Two or more copies of an edit source list may exist in order to differ.
[0113] The list of removal edits set up with the processing box 410 is based on RIMUBUDO links, and, on the other hand, the list of additional edits set up with the processing box 414 is based on ADIDDO links. In case a list is set up by the processing box 410 or 414, the walker routine 222 accesses each edit in a suitable edit source list, and receives the suitable entry about the list of [under the ****

setup for edit]. In each case, the edit in an edit source list is used, and the edit identifier which shows the node ID of the child node of the link of edit and the type of edit under activation comes to hand.

[0114] The parents of the child node are added after the list of nodes which does not already exist on a list and which was restricted and influenced. the parent node to which, as for parents, the child node was extended -- or the parent node of the child node -- current -- when nothing is extended, it is the parent node of the beginning of the child node. When related to the child node in which the edit from an edit source list does not have parents, the edit is placed after the list of nodes which must be related to the root therefore by which the root node was influenced in that case.

[0115] Before accessing the next edit in the suitable edit source list used for the last of the iteration about the link from a source edit list repeatedly [of a degree], the child node of the link is also added to a child node list. Thus, it is repeatedly carried out about each edit of an edit source list until all edits are processed, in order to complete the list of the nodes and child nodes which were influenced.

[0116] Next, in the processing box 412 or 416, edit is arranged using a list, a sequence similar to the sequence of the processing boxes 354-382 of drawing 7 about each node in the list of influenced nodes is followed, the node of software is pushed in front of a queue rather than a root node from a list, and some change is made as follows with the processing box 372. As for the layout of the processing boxes 412 and 416, in addition to discernment of which child is a non-extending leaf, each child determines whether exist in a child node list. When that is right, a layout carries out the multiplication of the include angle about the child, and the radius under the weight. In the processing box 412, since weight is 0, with the processing box 374, an include angle and a radius are mostly arranged by the location before that 0, therefore the child disappears. With the processing box 416, since weight is 1, the child is arranged by the new location in whenever [full-size], and a total radius in the processing box 374.

[0117] Actuation of the processing boxes 410-416 may also be carried out within an animation sequence, in that case, removal edit may be processed in the part of the beginning of an animation sequence, and additional edit may be processed in the consecutiveness part of a sequence. On the other hand, as mentioned above with regards to drawing 6 , when non-animation edit mainly arises from the node generation in paint, the edit may be only additional edit and the edit under all present holds may be processed at each step of an animation sequence.

[0118] As the demand which animates modification according to individual of a large

number which answered the insert/delete event like the demand which contracts or extends an element, or were demanded by such event is answered and it may be generated, when modification is animation edit, the walker routine 222 is the processing box 420, and the number of the elements first removed and added based on a source edit list is received. On the other hand, the number added can be obtained by adding the number of the elements in extractives pan dead links and ADIDDO links by the ability obtaining the number removed by adding the number of the elements in colla tempestade PUSUTO links and RIMUBUDO links. Next, with the processing box 422, the walker routine 222 assigns an usable animation step between a removal step and an additional step, and also sets up the list of removal edit and additional edits in it like the processing boxes 410 and 414 of drawing 8 a little. Although simple allocation of an animation step is a half removal step and a half additional step, when all steps can turn into an additional step when there is no element removed, and there is no element added, it is the reverse.

[0119] It sets to removal edit in the processing box 422, and the setup of an additional edit list, and unless the node contracted or extended exists, the walker routine 222 can be performed as mentioned above with regards to the processing boxes 410 and 414. In the case of the node contracted or extended, in addition to the parents, the node itself is pushed after the influenced node list, next the child of the node instead of the node itself is added to a child node list. In other words, unlike other actuation of influencing only one node, it can be considered that contraction or an escape influences the node of two generations. The walker routine 222 sets up the list of two pairs, the influenced node and a child node, and one pair is used for removal edit and it uses one more pair for additional edit.

[0120] next, the animation step which removes a node performs by the loop formation started with the decision box 430 -- having -- the processing box 432 -- weight -- receiving -- weight -- using -- removal edit -- arranging -- an animation frame -- mapping -- and paint is carried out. the animation step which similarly adds a node by the loop formation started with the decision box 440 next performs -- having -- the processing box 442 -- weight -- receiving -- weight -- using -- additional edit -- arranging -- an animation frame -- mapping -- and paint is carried out. Before adding a node, by removing a node, the situation that the same node appears in two places within a single frame is prevented. This technique can ensure that the last weight is 0 or 1, respectively by performing the one last step by weight 0, after removing a node, and performing another last step by weight 1, after adding a node.

[0121] By lengthening a current removal animation step number from the removal

animation number of steps with the processing box 432, and then dividing the difference by the removal animation number of steps, weight can come to hand so that weight may be set to 0 from 1 among a series of removal animation steps. Similarly, with the processing box 442, by adding 1 to a current additional animation step number, and then dividing the sum total by the additional animation number of steps, weight can come to hand so that weight may be set to 1 from 0 among a series of additional animation steps.

[0122] The total of an animation step helps the effect of the consciousness on object permanence animation on display with an animation rate. The total of an animation step determines indirectly the rate which must move with regards to the field of the element to which the rate from which the field of the element removed or added changes was determined, therefore other elements were removed or added so that he can understand from explanation of the approach of obtaining above-mentioned weight. The direction with much animation number of steps assigned [that sufficient animation rate is maintained and] appropriately between a premise, then a removal step and an additional step is suitable producing object permanence.

[0123] When the technique of drawing 8 was performed at a number with a suitable suitable rate of animation steps, it made well consciousness of 1 set of nodes contracted and extended like the fan which it can be closed a little or can be opened. By adjusting the radius and include angle which were assigned to the node, or the node which is pressed out to infinity or was inserted or the deleted node was drawn in the parents grows and comes out from the parents, different consciousness which is pulled from infinity can be obtained. When the child seems to be pressed out to infinity when only one child is deleted from one child's group, and all children are deleted as one group like contraction, all children are able to seem to be drawn in parents. When the child looks [pull / from infinity] the same when one child is added by one group, and all children are inserted as one group like an escape, all children are able to seem to grow and come out from parents. Furthermore, at the rate adjusted so that a grandchild might be stabilized, and might be seen and only a child might be moved and seen, while the child is won over to parents, a grandchild can be pressed out to infinity.

[0124] As mentioned above, it is possible to animate in parallel modification according to two or more individuals shown within an edit list. Furthermore, a client 212 can be equipped with an interface so that it can require that a change according to two or more individuals should be made without animation through an interface. In other words, an edit list can continue being formed until the command which requires completion of edit is received. A command can require that edit should be animation

as mentioned above, or should be completed without animation like refresh actuation. An edit list is eliminable if edit is completed.

[0125] In order to make animation easy, additional escape/contraction information is maintainable. For example, an additional flag can enable quick exchange with the expression of front structure, and the expression of the changed structure by enabling rapid decision about which element should be scanned about each expression.

[0126] Drawing 9 and 10 show the instantiation-sequence of an expression of node-link structure which is displayed by the operation as the present software, and are substantially [as the sequence mentioned above with regards to 8 from drawing 4] the same. The expression currently illustrated shows the organization chart and resembles the expression of 21 a little from drawing 17 of U.S. Pat. No. 5,619,632.

[0127] Drawing 9 shows the sequence of the expression which may be produced from the demand which contracts a node feature, and, on the other hand, drawing 10 shows the sequence of the expression which may be produced from the demand which extends a node feature. Moreover, the sequence connected through drawing 10 from drawing 9 may be produced from the demand which extends the clone of the node feature extended before so that it may explain below.

[0128] The expressions 500, 502, and 504 of drawing 9 contain the node feature showing the individual in an organization chart. When each node feature is displayed with the maximum size, it contains the rectangular title block which has identifier or nickname of the graphic expression of a person's face, and its person. there may be "small +" or small "-" for each to require an escape or contraction of a node feature in the bottom right corner of the title block of the rectangle of the node which has a descendant. These small symbols can change like [in the 1st animation frame] at the suitable time, after being chosen. If a node feature is displayed smaller than the maximum size, it can have the size to which the graphic expression was reduced, and a title block can be short-****(ed). If a node feature approaches an expressional periphery, it can become a small point and, subsequently can disappear.

[0129] The node feature 510 exists in each of expressions 500, 502, and 504, it is extended and shown by expression 500 and the child is shown with it by the small point near the periphery. The sequence of drawing 9 may be produced from the demand which contracts the node feature 510. This demand is answered, the child and grandchild of the node expressed by the node feature 510 are removed from node-link structure, and the transition to the expression with absent child and grandchild from expression 500 is animated. Expressions 502 and 504 are two middle expressions which may be displayed by such animation sequence.

[0130] In expression 502, the child of the node feature 510 just began to be lengthened to the node feature 510. However, since the child has spread from the child at the rate which offsets the rate lengthened toward the node feature 510, the grandchild of the node feature 510 stops at a predetermined location.
[0131] With expression 504, the child is further lengthened by near toward the node feature 510, and, on the other hand, a grandchild is continuing stopping at a predetermined location. Furthermore, since the child separated from the expressional periphery, he has sufficient field which can display the title block of each child's rectangle. The title block has produced the consciousness of the card or sheet which was able to be opened to overlap and 1 set of flabellate forms based on the order by which paint was carried out.
[0132] Since all children converge on the node feature 510, the pass which the child of the node feature 510 follows is not parallel. If a child approaches the node feature 510, the fields of the node feature 510 will decrease in number, and the size of a graphic expression a person's face will be reduced temporarily.
[0133] The last expression with which the child or grandchild of the node feature 510 is not displayed at all follows expression 504 after some middle expressions the back potentially. The node feature 510 can understand this situation from the expression 540 of drawing 10 which seems to be visible within the last expression.
[0134] At drawing 10 , the node feature 550 of expressions 540, 542, and 544 which is extended in the expression 544 of drawing 10 and shown although it contracts by the expression 504 of drawing 9 , respectively and is shown is included, and the child is shown by the small point near the periphery. The sequence of drawing 10 may be produced from the demand which extends the node feature 550, when the node feature 550 seems to be visible with expression 504. This demand is answered, the child and grandchild of the node expressed by the node feature 550 are added to node-link structure, and the transition to the expression with which a child and a grandchild exist from expression 504 is animated. Expressions 540 and 542 are two middle expressions which may be displayed by such animation sequence, and expression 544 is the last expression.
[0135] In expression 540, the child of the node feature 550 just began to spread from the node feature 550. Since only a certain distance has still separated the child from the expressional periphery, he has sufficient field which can display the title block of each child's rectangle. Like the expression 504 of drawing 9 , the title block has produced the consciousness of the card or sheet which was able to be opened to overlap and 1 set of flabellate forms. However, since a child is lengthened toward a

child from infinity at the rate which offsets the rate which spreads from the node feature 550, the grandchild of the node feature 550 is already in a predetermined location.

[0136] With expression 542, the child separated from the node feature 550 further, and has spread, and, on the other hand, a grandchild is continuing stopping at a predetermined location. Furthermore, since the child approached by the expressional periphery, he does not have sufficient field for displaying a title block, but has become a small point.

[0137] Finally by the expression 544, the child of the node feature 550 has arrived at the location near an expressional periphery. A grandchild is not seen according to it being close to a periphery.

[0138] Since all children diffuse the pass which the child of the node feature 550 follows from the node feature 550, it is not parallel. Since the child is close to the node feature 550 when a child appears first with expression 540, the fields of the node feature 550 decrease in number, consequently the size of a graphic expression a person's face is reduced temporarily. The field of the node feature 550 increases as a child distributes from the node feature 550.

[0139] As mentioned above, the single sequence to which the sequence of drawing 9 and drawing 10 is connected, the node feature 510 is contracted first, and then the node feature 550 is extended may be generated. Although the node features 510 and 550 express the same node in a graph, since the node has two close links, the clone of the node is generated, and this may be produced [both], when generating twice within the tree expressed with drawing 9 and 10. If a user demands the escape of the node feature showing one clone, the response can include contraction of the node feature showing the clone of another side extended before. In fact, the branch is moved to a node 550 from a node 510. "+" in two node features and the symbol of "−" are changed appropriately.

[0140] Although operation similar to above-mentioned operation was well performed on the processor compatible [PC] with IBM, operation may be performed with other equipments which have the suitable processor of arbitration.

[0141] Although operation similar to above-mentioned operation was well performed using C++ language in the 32-bit Windows (Windows) environment, it may use other programming language and environments including a non-object-oriented environment, and may use other platforms, such as LISP (Lisp), a UNIX (Unix) environment, ANSI C, and a pascal (Pascal).

[0142] although operation similar to above-mentioned operation was well performed

using the node-link data displayed in an XML conformity format and the existing experimental format, this invention is static or dynamic -- one of the inside of memory or networks are minded -- you may perform using the node-link data of accessible arbitration suitable type by the suitable approach of arbitration like.

[0143] Although it carried [use] out repeatedly [each / which operation similar to above-mentioned operation answers a navigation signal, and prepares and displays a series of one expression of a graph expressed or animated], this invention may be carried [use] out repeatedly [of the type of the others called by other signals or calls of a type].

[0144] Operation similar to above-mentioned operation was well performed using the navigation signal related to the expression of a single string expressed or animated with which it was received from the keyboard and the mouse, and node-link structure was displayed. However, this invention may be carried out, without using, using a navigation signal. For example, sorting from which the child of a certain node differs may be answered, or application of a different filter to the element of a certain structure may be answered, and you may make it move about an element. Moreover, this invention may be carried out using other navigation signals including the signal produced from selection of an item like the menu entry which requires an escape of the bottom of the shown enquiry of an escape and contraction signal of arbitration suitable type, or the exterior, node, or link, or selection of an item like the menu entry which requires an escape of the bottom of a current focus. A navigation signal may be related to space instead like the space generated by video game or the virtual reality environment which is not actually, or operating spaces other than a display, and a navigation signal may be instead generated by the suitable user input equipment containing the equipment of other classes for receiving a linguistic input like the pointing device of other classes and an alphabetic character or voice, and a gesture, or other user inputs of a format of arbitration. This invention may be carried out using the suitable animation technique of arbitration.

[0145] A node is expressed by features, such as an image of the shape of a circle, a rectangle, and an icon, and, as for above-mentioned operation, displays the expression of the node-link structure expressed by the line by which a link connects a node feature. However, this invention may be carried out about some modes of this invention using other suitable approaches of the arbitration expressing node-link structure including the operation in the expression including the hierarchy list of items which have the child of the indented item downward at least.

[0146] Although above-mentioned operation maintains the object permanence about

the feature which lets the sequence of a step pass and expresses a share node and a link by displaying a similar feature at the interval of time amount and space small enough, using other techniques for maintaining object permanence, this invention has the object permanence about the feature to which only a share node expresses only a share link, and may be carried out. For example, in order to help offer of the object permanence of a migration element, blurring between locations, the sequence of the profile between locations, or other queues like other indicators in which migration between locations is shown may be used.

[0147] The field for the feature which expresses with above-mentioned operation the element deleted or inserted decreases in number or increases, respectively, while the fields of a neighboring element increase or decrease in number. However, or this invention enters from a periphery, it may be carried out using other techniques of changing a feature showing the element deleted and inserted like other conventional animation techniques of arbitration including an usable technique, by PowerPoint (PowerPoint) of modification of the color which shows the migration feature, deletion, and insertion which appear in a periphery, fade-in, fade-out, or Microsoft Corp. (Microsoft). Furthermore, this invention may be carried out using the technique of increasing or decreasing fields other than the field of a neighboring element. Furthermore, it may be generated immediately and deletion may produce insertion without animation immediately similarly in the end of the animation with which migration of a neighboring element is compensated at the beginning of the animation with which migration of the neighboring element which may be animated is compensated without animation.

[0148] In above-mentioned operation, 1 set of elements are deleted, then 1 set of elements are inserted, and migration of an element deletes an element from a front location first, and is completed by next inserting an element in a new location. However, this invention may be carried out using the technique which deletes and inserts an element in parallel, and this invention may be further carried out using the technique which moves by the approach except deleting an element and inserting in a degree.

[0149] Although above-mentioned operation obtains layout data by a certain specific approach, this invention may be carried out by obtaining layout data by other approaches according [or] to carrying out, without using, using such a layout, or arranging the whole node-link structure according to an individual about each expression so that according to arranging node-link structure by other approaches.

[0150] Although paint is carried out in above-mentioned operation by the specific

approach which node-link structure is mapped by the unit disk and is in a degree This invention may be carried out without using, using mapping. Or include what node-link structure is mapped to other suitable rendering space of arbitration including three-dimension rendering space and an operating space, and is displayed on other suitable operating spaces of arbitration. You may carry out so that node-link structure may be mapped and displayed by other suitable approaches of arbitration.

[0151] Above-mentioned operation is suitable for the display of an expression of a tree. This invention may be used for displaying the expression of node-link structure of other types like a general graph.

[0152] Although above-mentioned operation uses the node-link data containing the extended flag of the link for defining the tree in a graph using the memory management performed by a certain specific approach, this invention may be carried out using the node-link structure which is defined by other suitable approaches of arbitration and loaded to memory by the suitable approach of arbitration.

[0153] although above-mentioned operation can treat the directed graph containing a circulation directed graph, this invention changes the link of other types into the suitable combination of a directed graph -- or another appearance -- if -- you may carry out for the graphs of other types by supplying the protocol for mapping the structure of a graph to a tree. For example, the **** link between two nodes may be changed into one pair of owner ** links between the same nodes, or may assign a direction based on suitable criteria. Although there is an inclination confused visually since each set of an owner ** link generally circulates through the expression from which all **** links are changed into one pair of owner ** links as a result, this derangement may be conquered by displaying circulation by the option.

[0154] Processing is performed by above-mentioned operation with the ranking which may be changed in many cases. For example, in drawing 6 , a depth-first round may be performed rather than a width-of-face priority round.

[0155] In the same above-mentioned operation, although the part of some software is distinguished by GURAFA, a walker, a painter and a mathematics routine, and the list like a client, this invention may be carried out by the software constituted from a suitable approach of arbitration by other combination lists of hardware and software.

[0156] This invention was applied to offer of the interactive browser of node-link structure. This invention may be applied to various contexts by which node-link structure is visualized. Especially this invention may be applied to visualization of web related structure like the structure formed of 1 set of web pages or other web objects which were stored in the cache.

[0157] More generally this invention may be applied to offer of the browser for an organization chart, a file system hierarchy, a hypertext hierarchy, World-Wide-Web connectability structure, parts decomposition, SGML structure, or other big node-link structures of arbitration. This browser may be used for edit of the contents of structure or structure.

[0158] Although this invention has been described with regards to the example as software, this invention may be carried out using exclusive hardware.

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the typical flow diagram showing the sequence of the node-link expression accompanied by modification.

[Drawing 2] It is the flow Fig. showing the general processing in the display of an expression sequence like the expression sequence of drawing 1 .

[Drawing 3] It is the typical diagram showing the common component of the equipment which displays an expression sequence like the expression sequence of drawing 1 .

[Drawing 4] It is the flow Fig. showing the general processing in the preparation which displays an expression sequence like the expression sequence of drawing 1 based on the signal which requires modification according to individual.

[Drawing 5] It is the typical diagram of a system.

[Drawing 6] It is the flow Fig. showing how an event can be answered when the system

of drawing 5 displays the expression of a directed graph.

[Drawing 7] It is the flow Fig. showing how the first layout can be performed by drawing 6 .

[Drawing 8] It is the flow Fig. showing whether the layout of the node-link structure changed how by drawing 6 can be performed.

[Drawing 9] It is drawing showing two sequences of an image generable [with the technique of drawing 8].

[Drawing 10] It is drawing showing two sequences of an image generable [with the technique of drawing 8].

[Description of Notations]

10 Display

20 1st Expression

40 Middle Expression

50 The Last Expression

212 Client

220 GURAFA Routine

222 Walker Routine

224 ** INTARUCHIN

226 Mathematics Routine

230 Directed Graph DS

232 Node Location Data

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2000－99226
（P2000－99226A）
(43)公開日　平成12年４月７日(2000.4.7)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)
Ｇ０６Ｆ	3/00	６５２	Ｇ０６Ｆ	3/00	６５２Ｚ
	17/30			15/403	３７０Ｚ
				15/419	３１０

審査請求　未請求　請求項の数１　ＯＬ　（全 23 頁）

(21)出願番号　特願平11－214490

(22)出願日　平成11年７月29日(1999.7.29)

(31)優先権主張番号　１２４５２８

(32)優先日　平成10年７月29日(1998.7.29)

(33)優先権主張国　米国（ＵＳ）

(71)出願人　590000798
ゼロックス　コーポレイション
ＸＥＲＯＸ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　06904－1600　コネティカット州・スタンフォード・ロング　リッチ　ロード・800

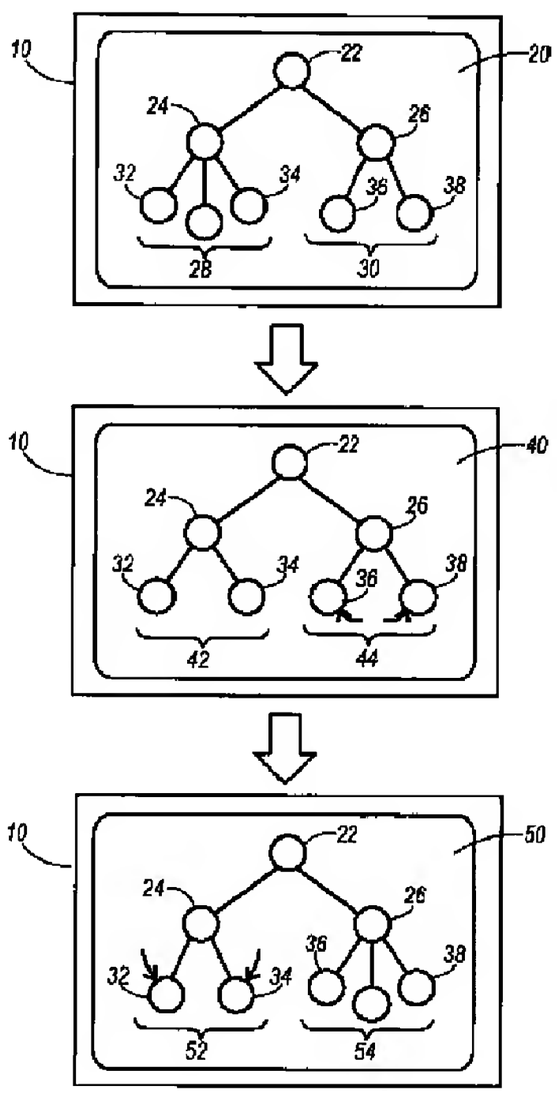
(72)発明者　ティチョミア　ジー．テネヴ
アメリカ合衆国　95123　カリフォルニア州　サンノゼ　クリー　ドライブ　610

(74)代理人　100079049
弁理士　中島　淳　（外１名）

最終頁に続く

(54)【発明の名称】　ディスプレイ上にノードーリンク表現を表示する方法

(57)【要約】
【課題】　変更を伴うノードーリンク構造のディスプレイ表示方法の提供。
【解決手段】　方法は第１ステップ、最終ステップ、及び少なくとも１中間ステップを含むステップのシーケンスから成り、第１ステップは第１ノードーリンク構造を表す第１表現２０を表示し、最終ステップは変更後の第２ノードーリンク構造を表す最終表現５０を表示し、第１及び第２ノードーリンク構造の共有要素は第１表現と最終表現とで位置が異なるフィーチャ３２、３４、３６及び３８で表される移動要素を含み、中間ステップは個々に移動要素３２、３４、３６及び３８を含む共有要素のサブセットを表すフィーチャを含む中間表現４０を表示し、個々のサブセットも第１及び最終表現内のフィーチャで表され、移動要素を表すフィーチャ３２、３４、３６及び３８はステップのシーケンスを通してオブジェクト不変性を有する。



1

【特許請求の範囲】
【請求項１】　ディスプレイ上にノード－リンク表現を表示する方法であって、該方法は第１ステップ、最終ステップ、及び少なくとも１つの中間ステップを含むステップのシーケンスから成り、
第１ステップは第１ノード－リンク構造を表わす第１表現を表示し、
最終ステップは第１ノード－リンク構造内での少なくとも１つの挿入及び少なくとも１つの削除によって変更されたバージョンである第２ノード－リンク構造を表わす最終表現を表示し、第１及び第２ノード－リンク構造は共に１組の共有要素を含み、共有要素は第１及び最終表現において異なる位置を有するフィーチャによって表わされる移動要素を含み、
中間ステップは個々に移動要素を含む共有要素のサブセットを表わすフィーチャを含む中間表現を表示し、個々のサブセットも第１及び最終表現内のフィーチャによって表わされ、
移動要素を表わすフィーチャがステップのシーケンスを通してオブジェクト不変性を有する、
ディスプレイ上にノード－リンク表現を表示する方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、変更が生じるノード－リンク構造の表示に関する。
【０００２】
【従来の技術】ジャーナル・オブ・ビジュアルランゲージズ・アンド・コンピューティング誌(Journal of Visual Languages and Computing)、1996年、第７巻、33頁－55頁の、ランピング(Lamping, J.)及びラオ(Rao, R.)の「双曲ブラウザ：大きな階層を視覚化するための焦点＋コンテキスト技術(“The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies”)」は、構成要素が外側に移動するにつれて構成要素のサイズが減少し、半径を増大させながら構成要素の数が指数的に増える、階層を表示するための技術を開示している。表示を滑らかに変形して、中心又は焦点でノードを変化させられる。異なるビューの間のアニメーション化された遷移はオブジェクトの不変性を維持し、ユーザが変化に順応するのを助ける。素早い再表示のために、フリンジをより少なく描画し、弧の代わりに線を描画することができ、アニメーション表示中はテキストを脱落させることができる。ランピングら(Lamping et al.)の米国特許第5,619,632号は、ノード－リンク構造を表示するための類似の技術を用いるブラウザを開示しており、構造の編集にブラウザを用いてもよいことに言及している。
【０００３】システムズ･アンド･コンピューターズ･イン･ジャパン誌(Systems and Computers in Japan)、1993年、第24巻、第8号、35頁－46頁のマツウラ(Matsuura,

2

T.)、タニグチ(Taniguchi, K.)、マスダ(Masuda, S.)及びナカムラ(Nakamura, T.)の「ブラウジング及びズーム能力を有する大きなツリーのためのグラフエディタ(“A Graph Editor for Large Trees with Browsing and Zooming Capabilities”)」は、ディスプレイ上での大きなツリーの直接操作を可能にするライブラリプログラムを開示している。ノード又はサブツリーが追加又は削除されると、データ構造が更新される。つまり、グラフレイアウトアルゴリズムに基づいて各ノードについてのレイアウトが計算され、各ノード及びエッジが再描画される。ノードが連続して追加又は削除されると、全ての変更がなされた後に、表示のただ１回の更新を行うことができる。ビューポートはグローバルビュー及びキャンバスを含み、更新は、自動更新、部分更新、又はオンデマンド更新である得るモードによる。部分更新はオンデマンド更新と似ているが、中間表示ノードが一時的に表示される。
【０００４】本発明は、変化するノード－リンク構造の表示における問題を扱う。
【０００５】ノード－リンク構造は幾つかの状況で変化し得る。例えば、構造が大きい場合は、所与の時間に入手可能なデータは構造の一部を定義するだけかもしれない。これは、完全な構造を定義するデータが未到着であること、あるいは構造が非常に大きくメモリが構造の完全な定義を保持するのに十分に大きくないために、生じ得る。そのような場合は、メモリに部分的な構造が保持され、追加データが到着した又は取り出された時に変更される。
【０００６】ノード－リンク構造が変化する別の状況は、構造が、基礎をなす構造の部分バージョンである場合である。例えば、ユーザは、フィルターにかけたバージョンを入手するために、特定のタイプのノード又はリンクを選択するように要求することができる。それに応答して、フィルタ基準によって選択されないノード及びリンクを省略するノード－リンク構造が生成され、対象のノード及びリンクにより大きいスペースを与える。或いは、構造は基礎をなす有向グラフ（ＤＧ）構造を表わすツリーであり得、有向グラフ構造内には、更に子を有するＤＧ内のノードへの複数の入リンクが存在する。１つの方法では、ＤＧノードは、ツリー内の、各々が入リンクの１つを有する複数の親ノードによって表わされるが、ＤＧノードの子はツリー内の親ノードの１つだけで代表される。ユーザの入力に応答して、子が代表される親ノードが変化することがあり、それに従ってツリー構造が変化する（しかしＤＧは変化しない）。
【０００７】また、基礎構造自体が変化することもある。例えば、基礎構造は、その基礎構造を見ているブラウザ以外のプロセスに応答して変更されることがある。或いは、基礎構造はブラウザインタフェースを介したユーザの要求に応答して変更されることがある。

【０００８】任意のこれらの方法で変化するノード－リンク構造を、本明細書では“動的ノード－リンク構造”と呼ぶ。従って、この用語は、変化する基礎ノード－リンク構造だけではなく、変化する、又はそれ自体は静的又は動的いずれでもよい基礎ノード－リンク構造から異なる方法で得られる部分バージョンも含む。

【０００９】ランピング及びラオによって述べられている表示技術及び他の従来の技術は、表示中の構造変化に伴う問題に導き得る。典型的に、少なくとも変化した構造の一部について新たなレイアウトがなされなくてはならず、次にそれが再表示されなければならない。従って、そのような技術を用いて表示中の構造内に変化が生じると、変化した構造の表現にしばしば唐突又は不連続な遷移が存在する。幾つかの場合には、レイアウト及び再表示が非常に遅く、ユーザが変化した構造と効果的に対話できない。更に、レイアウト及び再表示が、ユーザの遷移に対する理解を向上させるであろう古い構造から新しい構造への遷移のアニメーション化の方法を提供しない。従って、従来の表示技術の多くは、静的ノード－リンク構造に対して満足な結果を生成するだけであり、動的ノード－リンク構造には適していない。

【００１０】

【発明が解決しようとする課題】しかしながら、従来の表示技術の幾つかは、階層リストの項目を拡張又は収縮する遷移をアニメーション化できる。しかしながら、これらの技術は拡張又は収縮いずれか１つの変更に限定されており、従って、全ての遷移に対して汎用化するのは困難である。

【００１１】

【課題を解決するための手段】本発明は、ユーザがより広い範囲の遷移を理解するのを容易にする技術を提供することによって、動的ノード－リンク構造の表示における問題を緩和する。各技術は表現のシーケンスを表示し、最初の表現は第１ノード－リンク構造を表わし、最後の表現は第１ノード－リンク構造の変更バージョンである第２ノード－リンク構造を表わす。最初と最後の構造によって共有される要素は、最初の表現と最後の表現とでは異なる位置を有するフィーチャ(feature)によって表わされる移動要素を含む。この技術は、最初の表現と最後の表現との間に少なくとも１つの中間表現を表示する。このシーケンス中の各表現は、移動要素を含む共有要素のサブセットを表わすフィーチャを含む。移動要素を表わすフィーチャは、ステップのシーケンスを通してオブジェクト不変性を有する。

【００１２】本発明の１つの態様では、第２ノード－リンク構造は、第１ノード－リンク構造内の少なくとも１つの挿入及び少なくとも１つの削除を伴う変更バージョンである。例えば、要素は１つの位置から削除されて別の位置に挿入され得る。従って、この態様に従った、本明細書で“削除及び挿入”と呼ぶ技術は、要素を移動するために、又は複数の削除及び挿入を含むより複雑な変更を行うために用いられてもよい。

【００１３】新たな削除及び挿入技術はそれぞれ、削除又は挿入中の要素の領域が、連続画像の個々において減少又は増大いずれかの変化をする、画像のシーケンスを表示する。同時に、近隣要素に補償領域変化が生じる。領域変化が適切に選択された場合は、画像のシーケンスは、削除においては近隣要素が削除中の要素に接近し、挿入においては挿入中の要素が近隣要素を脇に押しやる印象を生み出す。

【００１４】画像のシーケンスは、それぞれの場合に、削除又は挿入中の要素の適切な動きを提供することもできる。例えば、楔形の領域を有する要素の削除では、削除中の要素はその楔の開いた端部から出ることができるか、その楔に沿って縮むことができるか、又はその楔が縮み始める前に消えることができる。

【００１５】本発明の別の態様は、２次元ディスプレイでの遷移をアニメーション化する技術の発見に基づいている。この技術に従って、２つの移動要素は平行な直線ではないパスを辿る。例えば、要素は曲線パスに沿って移動してもよく、独立した平行ではないパスに沿って移動してもよい。

【００１６】本発明の更に別の態様は、ノード－リンク構造の個別の変更を並行してアニメーション化する技術の発見に基づいている。この技術に従って、少なくとも２つの個別の変更を要求する信号が受信され、この信号に基づいて要求された変更を示す変更データが入手される。次に、変更データ及びノード－リンク構造を定義する第１データ構造を用いて、ノード－リンク構造の変更バージョンを定義する第２データ構造を入手し、それに従って表現を表示できる。

【００１７】この新技術は、ユーザが動的ノード－リンク構造内の遷移を理解するのを容易にするので、有益である。更に、この新技術は、完全なノード－リンク構造のレイアウト及び表示の実行に必要な時間のかかる計算をせず、代わりに増分レイアウト及び再表示を用いて実施できる。その結果、ユーザはより効果的に動的ノード－リンク構造と対話できる。

【００１８】

【発明の実施の形態】以下の概念的構想は、本明細書に参照として援用する米国特許第5,590,250号及び第5,619,632号で述べられた概念的構想と共に読むと、本発明の広い範囲を理解する上で役に立ち、以下に定義される用語は、特許請求の範囲を含む本出願を通して示された意味を有する。

【００１９】“ノード－リンク構造”は、ノードとリンクとに区別できる項目を含み、各リンクが２つ以上のノードに関係している構造である。“グラフ”は、各リンクが２つのノードに関係しているノード－リンク構造である。“有向グラフ”は、各リンクが関係するノード間

5

の方向を示し、一方のノードがリンクのソース又は“フロムノード”であり、他方のノードがリンクの宛先又は“ツーノード”であるグラフである。“非環式有向グラフ”は、リンクが、それらの示された方向に辿られたときに、任意のノードからそれ自体に戻るパスを提供しない有向グラフである。“ツリー”は、ツリー内の任意の非ルートノードについて、リンクが、その示された方向に辿られたときに、ルートノードで始まり非ルートノードへと導く１つのパスだけを提供するような、１つのルートノードだけを有する非環式有向グラフである。

【００２０】ノード－リンク構造の“要素”は、ノード－リンク構造のノード及びリンクである。

【００２１】“ノード－リンク表現”は、ノード－リンク構造の表現である。例えば、リンクはリンクフィーチャによって表わすことができ、ノードはノードフィーチャによって表わすことができる。

【００２２】要素を表わすあるフィーチャが、表示されるディスプレイの各次元の空間的範囲を有する場合、そのフィーチャは“領域を有する”。例えば２次元ディスプレイ上では、フィーチャがディスプレイの両方の次元に延びる場合、そのフィーチャは領域を有する。

【００２３】ノード－リンク表現内のある要素を表わすフィーチャの“近隣フィーチャ”は、他の要素を表わすフィーチャを含み、近隣フィーチャは、あるフィーチャと他の要素を表わす他のフィーチャとの間の距離の中央値よりも、あるフィーチャにより近接している。

【００２４】シーケンス中の表現内の要素を表わすあるフィーチャが、全て同一フィーチャであるように見える場合、ノード－リンク構造の要素を表わすそのフィーチャは、ノード－リンク構造の表現のシーケンスを通して“オブジェクト不変性を有する”。例えば、シーケンス中の任意の２つの連続した表現内のフィーチャは、それらが同一フィーチャに見えるように十分類似し且つ時間及び空間的に十分近接していてよい。オブジェクトの不変性を有するフィーチャは“パスを辿って”移動するように見える。

【００２５】“アニメーションループ”は、各反復が画像を表示する反復動作であり、各反復において、各画像内のフィーチャが、オブジェクト不変性によるように、次に続く先行画像内のフィーチャの継続のように見える。“アニメーションサイクル”は、アニメーションループの１回の反復である。

【００２６】ある要素が、ノード－リンク表現内に少なくとも１つの子を有する場合、その要素はノード－リンク表現内で“子孫を有する”又は“拡張されている”。ある要素が表現内に全く子を持たない場合、その要素は“子孫を持たない”。ある要素が、基礎をなすノード－リンク構造に基づき得るにもかかわらず、表現内に全く子孫を持たない場合、その要素は“収縮されている”。

【００２７】本明細書では、“ナビゲーション信号”という用語は、ユーザがノード－リンク構造のある部分に他の部分よりも高い興味を持っていることを示す信号を意味する。例えば“拡張信号”は、グラフのある要素の表現が拡張されているグラフ表現を表示する要求を示し、一方“収縮信号”は、グラフのある要素の表現が収縮されているグラフ表現を表示する要求を示す。他の例は、ノード－リンク構造の一部を特定の位置で表示する要求を含み、それはブックマーク等を選択することによって、又は指し示されたフィーチャを焦点の中心へと移動することを要求するポイント及びクリック動作によって可能である。

【００２８】１つ以上の要素の挿入又は削除のように、又は挿入及び削除の組合せによって実施し得る移動又はコピー動作のように、ある信号が構造内の１つ以上の要素の変更を要求する場合は、その信号はノード－リンク構造内の“変更を要求する”。拡張及び収縮信号は、変更を要求する信号の例である。

【００２９】２つの変更が各々他方から独立して行われ得る場合は、２つの変更は“個別である”。

【００３０】図１では、ディスプレイ１０が、第１ノード－リンク構造の第１表現２０で開始し、第１ノード－リンク構造に、２つの個別の変更を要求する信号から生じ得る削除及び挿入の両方を行った変更バージョンである第２ノード－リンク構造の最終表現５０で終了する、ノード－リンク表現のシーケンスを表示している。ディスプレイ１０は表現２０と５０との間に中間表現４０を表示する。

【００３１】第１ノード－リンク構造は、フィーチャ２２によって表わされる第１水準ノードを有し、第１水準ノードはフィーチャ２４及び２６によって表わされる第２水準ノードである２つの子を有する。フィーチャ２４によって表わされる子はブランチ２８の頂点にあり、ブランチ２８は、第３水準ノードである３つの子を含む全ての子孫を含む。同様に、フィーチャ２６によって表わされる子はブランチ３０の頂点にあり、ブランチ３０は、第３水準ノードである２つの子を含む全ての子孫を含む。

【００３２】最終表現５０によって示されている第２ノード－リンク構造は、多くの要素を第１ノード－リンク構造と共有しており、そのサブセットは表現２０、４０、及び５０全ての内のフィーチャによって表わされる。共有要素は、フィーチャ２２、２４、及び２６によって表わされるノードと、それらを結ぶリンクとを含み、各々が表現２０、４０、及び５０を通して安定した位置を有する。同じく共有要素は、ブランチ２８内のフィーチャ３２及び３４によって表わされるノードと、ブランチ３０内のフィーチャ３６及び３８によって表わされるノードと、ノード３２、３４、３６、及び３８へのリンクを表わすフィーチャとを含み、これらのノード及びリンクのうちの１つを表わす各フィーチャは、最終表

現５０内では第１表現２０内とは異なる位置を有し、従って、これらのノード及びリンクを本明細書では“移動要素”と呼ぶ。

【００３３】図１に示されている例では、移動要素を表わすフィーチャの位置の変化は、第１ノード－リンク構造が第２ノード－リンク構造を得るために２つの点で変更されたことを理由として生じる。ブランチ５２を得るために、ブランチ２８内で表わされている第３水準ノードのうちの１つが削除され、ブランチ５４を得るために、ブランチ３０に追加の第３水準ノードが付加されている。第１及び第２ノード－リンク構造間の変更並びにその結果生じた位置の変化に関わらず、移動要素を表わすフィーチャは表現のシーケンスを通してオブジェクト不変性を有する。表現４０及び５０内の矢印によって示されるように、ノード３６及び３８は曲線パスを辿って離れるように見え、一方、ノード３２及び３４は互いに向かう曲線パスを辿るように見える。従って、これらのノードは平行な直線ではないパスを辿るように見える。

【００３４】シーケンスを通してのオブジェクト不変性は様々な方法で生み出し得るが、図１に示されている技術は、空間及び時間の小さなインターバルで類似のフィーチャを表示することによってオブジェクト不変性を生み出す例を示している。フィーチャ２２、２４、及び２６、並びにそれらの間のリンクを表わすフィーチャの各々は、同じ位置で素早く続けて表示されている間、実質的に同一であり続けることが可能であり、オブジェクト不変性を確保する。フィーチャ３２、３４、３６、及び３８、並びにそれらの間のリンクを表わすフィーチャの各々も、僅かに変位した位置で素早く続けて表示されている間、実質的に同一であり続けることが可能である。フィーチャ３２及び３４の位置の変位は、それらを互いに近づけ、フィーチャ２４の子である第３水準ノードの削除を補う。即ち、フィーチャ３２及び３４は、例示するように、ノードが削除されたブランチ５２内では、削除前のブランチ２８内及び最初に削除が行われたブランチ４２内より、互いに近接している。フィーチャ３６及び３８の変位は、フィーチャ２６の子である第３水準ノードの挿入を補うために、フィーチャ３６及び３８を離間移動する。即ち、フィーチャ３６及び３８は、ブランチ４４内ではブランチ３０内よりも更に離間しており、ブランチ５４内で第３水準ノードが挿入されている。

【００３５】図２では、処理ボックス１００の処理は、第１表現を表示することによって開始する。第１表現は第１ノード－リンク構造を表わし、第２ノード－リンク構造と共有の要素のサブセットを表わすフィーチャを含む。サブセットは、図１のフィーチャ３２、３４、３６、及び３８のうちの１つで表わされるノードのような移動要素を含む。

【００３６】次に、処理ボックス１０２の処理は、同じく共有要素のサブセットを表わすフィーチャを含む中間表現を表示する。点線で示されるように、処理ボックス１０２内の処理は、一連の２つ以上の中間表現を表示するために、１回以上実行されてもよい。各中間表現内では、移動要素を表わす各フィーチャは、第１表現又は先行する中間表現であり得る先行表現内の移動要素を表わすフィーチャとのオブジェクト不変性を有する。

【００３７】最後に、処理ボックス１０４の処理は、第１ノード－リンク構造の変更バージョンである第２ノード－リンク構造を表わす最終表現を表示する。処理ボックス１０２のように、最終表現は共有要素のサブセットを表わすフィーチャを含み、移動要素を表わすフィーチャは、第１表現から最終表現在のへの位置の変化に関わらず、先行する中間表現内の移動要素を表わすフィーチャとのオブジェクト不変性を有する。

【００３８】図３の装置１５０は、ユーザ入力回路１５４からユーザ信号を示すデータを受信すると共にディスプレイ１５６に画像を定義するデータを供給するために接続された、プロセッサ１５２を含む。プロセッサ１５２は、第２構造が第１構造の変更バージョンである第１及び第２ノード－リンク構造を定義するノード－リンクデータ１５８にアクセスするためにも接続されている。プロセッサ１５２は、例えば、メモリ１６４、記憶媒体アクセス装置１６６、又はネットワーク１６８への接続から受信した命令を供給できる命令入力回路１６２を介して、命令を示す命令データ１６０を受信するためにも接続されている。

【００３９】命令データ１６０よって示される命令の実行において、プロセッサ１５２はディスプレイ１５６に、第２ノード－リンク構造と共有の要素のサブセットを表わすフィーチャを含む第１ノード－リンク構造の第１表現を表示させる。共有要素のサブセットは上述のように移動要素を含む。次に、プロセッサ１５２はディスプレイ１５６に、同じく各々が共有要素のサブセットを表わすフィーチャを含む少なくとも１つの中間表現を表示させる。即ち、移動要素を表わすフィーチャは、先行表現内の同一共有要素を表わすフィーチャとのオブジェクト不変性を有する。最後に、プロセッサ１５２はディスプレイ１５６に、同じく共有要素のサブセットを表わすフィーチャを含む第２ノード－リンク構造の最終表現を表示させ、移動要素を表わすフィーチャは、先行する中間表現内の移動要素を表わすフィーチャとのオブジェクト不変性を有する。

【００４０】上述のように、図３は、命令入力回路１６２が命令を示すデータを受信し得る３つの可能なソース、即ちメモリ１６４、記憶媒体アクセス装置１６６、及びネットワーク１６８を示している。

【００４１】メモリ１６４は、ランダムアクセスメモリ（ＲＡＭ）又は読取り専用メモリ（ＲＯＭ）を含む、装置１５０内の任意の従来のメモリか、又は任意の種類の周辺又は遠隔メモリ装置であってよい。

【００４２】記憶媒体アクセス装置１６６は、例えば１組の１つ以上のテープ、ディスケット、又はフロッピーディスクのような磁気媒体、１組の１つ以上のＣＤ－ＲＯＭのような光学媒体、又はデータを格納するための任意の他の適切な媒体であり得る記憶媒体１７０にアクセスするための、ドライブ又は他の適切な装置又は回路であってよい。記憶媒体１７０は、装置１５０の一部、サーバの一部、他の周辺又は遠隔メモリ装置、又はソフトウェア製品であってよい。これらの場合のそれぞれで、記憶媒体１７０は装置１５０内に用いることができる１つの製造品である。記憶媒体１７０の上にデータユニットを配置して、記憶媒体アクセス装置１６６がデータユニットにアクセスしてそれらを命令入力回路１６２を介してプロセッサ１５２にシーケンスで供給できるようにすることが可能である。データユニットは、シーケンスで供給されると、図示されるように命令を示す命令データ１６０を形成する。

【００４３】ネットワーク１６８は、装置１８０から受信した命令データ１６０を供給できる。装置１８０内のプロセッサ１８２は、ネットワーク１６８を渡ってネットワーク接続回路１８４及び命令入力回路１６２を介して、プロセッサ１５２との接続を確立できる。どちらのプロセッサが接続を開始してもよく、接続は任意の適切なプロトコルで確立されてよい。次に、プロセッサ１８２はメモリ１８６に格納されている命令データにアクセスして命令データをネットワーク１６８を渡ってプロセッサ１５２に転送できるので、プロセッサ１５２はネットワーク１６８から命令データ１６０を受信できる。次に、命令データ１６０は、プロセッサ１５２によってメモリ１６４又はどこかに格納され、実行されることが可能である。

【００４４】図４は、どのように個別の変更を並行して処理できるかを示している。

【００４５】処理ボックス１９０の処理は、第１ノード－リンク構造内の少なくとも２つの個別の変更を要求する信号を受信する。例えば、信号は個別の変更を要求する２つの異なる信号を含んでもよい。

【００４６】この信号に基づいて、処理ボックス１９２の処理は、信号によって要求された変更を示す変更データを入手する。変更データは、使用に応じた適切なデータ構造で格納されてもよい。

【００４７】次に、処理ボックス１９４の処理は、変更データ、及び第１ノード－リンク構造を定義する第１データ構造を用いて、第２ノード－リンク構造を定義する第２データ構造を入手する。第２ノード－リンク構造は、第１ノード－リンク構造に要求された変更が実行されたバージョンである。

【００４８】最後に、処理ボックス１９６の処理は、第１及び第２ノード－リンク構造の表現を、アニメーションを伴って表示する。この処理は、図２に関係して上述された処理を含むことができる。

【００４９】上述した全般的なフィーチャは、ノード－リンク表現を表示するための多くの方法で様々な装置上で実施されてよい。以下に述べる実施例は、マイクロソフトウインドウズ(Microsoft Windows)の32ビット版を走らせ、C++言語ソースコードからコンパイルされたコードを実行する、ＰＣに基づくシステム上で実施されたものである。

【００５０】図５では、システム２００は、画像を表示するためのディスプレイ２０４、及びユーザから信号を供給するためのマウス２０８及びキーボード２０６に接続された、ＰＣプロセッサ２０２を含む。ＰＣプロセッサ２０２は、メモリ２１０及びクライアント２１２にアクセスできるようにも接続されている。メモリ２１０は、例示されているように、プログラムメモリ２１４及びデータメモリ２１６を含むことができる。クライアント２１２は、メモリ２１０に格納されたルーチン及びデータの組合せか、又は示されるようにメモリ２１０から独立していてよい、有向グラフに関する情報のソースである。例えば、プロセッサ２０２はネットワークを介してクライアント２１２と通信してもよい。

【００５１】プログラムメモリ２１４に格納されているルーチンは、幾つかの機能にグループ化できる。グラファルーチン２２０は、クライアント２１２からの情報によって定義される有向グラフを表わすデータ構造を生成及び変更する。ウォーカルーチン２２２は、有向グラフデータ構造から情報を入手することによって、キーボード２０６及びマウス２０８からのナビゲーション信号及び他のユーザ信号に応答する。ペインタルーチン２２４は、ディスプレイ２０４に信号を提供して有向グラフデータ構造の表現を表示させる。数学ルーチン２２６は、レイアウト空間内の有向グラフの要素の位置を得るために呼出されることができる。

【００５２】次に、データメモリ２１６は、プログラムメモリ２１４内のルーチンの実行中にプロセッサ２０２によってアクセスされるデータ構造を収容する。有向グラフデータ構造２３０は、上述のように、グラファルーチン２２０によって生成及び変更され得ると共に、ウォーカルーチン２２２及びペインタルーチン２２４によってアクセスされ得る。

【００５３】有向グラフデータ構造２３０内にリンクされ得る又は含まれ得るノード位置データ２３２は、双曲平面のような負曲空間内、及び２次元ユニットディスク（単位円）のようなレンダリング空間内のノードの位置を含むことができる。ノード位置データ２３２はプログラムメモリ２１４内のルーチンによってアクセスされ得る。

【００５４】プログラムメモリ２１４内のルーチンは、種々の雑データ構造２３４にもアクセスできる。データ構造２３４は、例えば、標準ヒープとして実施される、

１対のノードＩＤからリンクＩＤへのマッピングのための予備データ構造を含んでもよい。即ち、この予備データ構造は、一定の期待時間内のリンクＩＤの検索及び挿入を可能にする。

【００５５】図６は、どのように図５のシステムがグラフの表現を表示することによってイベントに応答できるかを示している。

【００５６】処理ボックス３００では、クライアント２１２は、開始グラフを入手すると共に、例えばノード生成の呼出しを介して、要素の初期セットをメモリにロードすることによって開始する。拡張フラグは要素の初期セット内のツリーを定義する。クライアント２１２は、双曲平面にツリーをレイアウトし、ツリーを双曲平面からユニットディスクへと、ルートノードをディスクの中心においてマッピングし、マッピングされたツリーをペイントし、二重バッファをスワッピングすることによってペイントされたバージョンをディスプレイ２０４に表示することを、全て処理ボックス３００で行うために、メモリ２１４内のルーチンに対する適切な呼出しも行う。

【００５７】処理ボックス３０２では、クライアント２１２はグラフに関係するイベントを受信する。イベントはナビゲーション信号、編集信号、又はユーザからの他のタイプの信号から生じ得る。或いは、イベントはシステム２００の内部又は外部いずれかの他のソースから受信されてもよい。いずれにせよ、イベントはクライアント２１２内から、メモリ２１４内のルーチンの１つから、又はプロセッサ２０２によって実行される他の命令からの呼出しの形態をとり得る。一連の受信されたイベントは、処理ボックス３０２がキューからのイベントの取り出しを含み得るように、キューの中に保持され得る。

【００５８】処理ボックス３０２で受信されたイベントに応答して、クライアント２１２は、メモリ２１４内のルーチンに対して１つ以上の呼出しを行うことによって、適切な応答を開始する。判断ボックス３０４によって示されるように、応答はイベントのタイプによるので、イベントに基づいてブランチが選択される。

【００５９】イベントは、オリエンテーション転換イベント、ストレッチイベント、又はドラッグイベントのような、非アニメーションイベントであってもよい。オリエンテーションイベントは、ユーザがルートノードに対して新たなオリエンテーションを示したときに生じ得る。ストレッチイベントは、ユーザが表示されている表現に対して新たなストレッチファクタを示したときに生じ得る。ドラッグイベントは、例えば、ユーザが、マウスダウン（マウスボタンを押したまま）クリック等によって表現内の位置を選択し、適切なジェスチャ又は他の信号によってその位置の移動を要求したときに生じ得る。

【００６０】処理ボックス３１０では、クライアント２１２は、イベントへの応答に必要な任意の情報を入手することによって、非アニメーションイベントに対する応答を開始する。オリエンテーションイベントについては、処理ボックス３１０で入手された情報が新たなオリエンテーションを含むことができる。ストレッチイベントについては、処理ボックス３１０で入手された情報が新たなストレッチファクタを含むことができる。

【００６１】ドラッグイベントについては、処理ボックス３１０での情報の入手は幾分複雑さが増す。クライアント２１２は、選択位置から最も近いノードのノード識別子（ノードＩＤ）を入手してもよく、また、要求された動きに関する情報を入手してもよい。情報のこれらの項目は、本明細書に参照として援用する米国特許第5,590,250号のコラム７１－７２及び図１４に関係して記載されている“最も近いノードを探索”という関数（機能）によって説明されているのとほぼ同じ方法で入手してもよい。

【００６２】クライアント２１２が処理ボックス３１０で必要な情報を入手したら、クライアント２１２はレイアウト、マッピング及びペイントのためのウォーカルーチン２２２及びペインタルーチン２２４に対する適切な呼出しで締めくくることができる。オリエンテーションイベントについては、ルートノードは新たなオリエンテーションでレイアウトされなくてはならない。ストレッチイベント又はドラッグイベントについてはレイアウトは必要ない。しかしながら、ストレッチイベントについては、ウォーカルーチン２２２に対する呼び出しが、マッピングに用いるための新たなストレッチファクタを含まなければならない。同様に、ドラッグイベントについては、ウォーカルーチン２２２に対する呼び出しが、マッピングに用いるための最も近いノードのノードＩＤ及び動きのパスに沿った次の位置を含まなければならない。

【００６３】処理ボックス３１２では、ウォーカルーチン２２２は、まず、双曲平面内に任意の必要なレイアウトを行ってもよく、また、ツリーの任意の保留中の編集をレイアウトしてもよい。次に、処理ボックス３１４では、ウォーカルーチン２２２は、開始ノードを開始位置において開始して、ツリーをユニットディスクにマッピングしてもよい。例えば、ドラッグイベントに応答して、開始ノードは処理ボックス３１０で識別された最も近いノードであってよく、開始位置は動きのパスに沿った次の位置であってよい。前にマッピングに用いられた開始ノード及び開始位置を、オリエンテーション又はストレッチイベントに応答して用いてもよい。

【００６４】ツリーがマッピングされたら、処理ボックス３１６で、マッピングされたツリーを表示バッファにペイントするためにペインタルーチン２２４を呼出すことができる。ペイント中、ペインタルーチン２２４はノ

ード生成の結果ツリー内で生じる新たな編集をマーキングできる。各編集は、フラグを設定することによって又は他の適切なデータを格納することによってマーキングできる。ペイントが完了すると、ペイントされたようにツリーを表示するために表示バッファのスワッピングを行うことができ、グラフの表現が供給される。

【００６５】上述のように、これらのイベントは現在は非アニメーションイベントとして実施される。オリエンテーションイベントに応答して、表現は新たなオリエンテーションに、典型的には表示領域の焦点にあるノードの周りを、旋回する。同様に、ストレッチイベントに応答して、表現は半径方向に、典型的には焦点にあるノードの周りで、拡張又は収縮する。ドラッグイベントに応答して、表現は入力信号によって決定されたレートで移動する。しかしながら、クライアント２１２は、オリエンテーションイベント、ストレッチイベント、又はドラッグイベントに対して、要求された変更を同等のより小さなイベントのシーケンスに変換し、処理ボックス３１０で、１つの呼出しが個々のより小さなイベントに対する一連の呼出しを発することによって、アニメーション化された応答を供給してもよい。

【００６６】図６は、現在の実施では常にアニメーション化して扱われる２つの異なるタイプのイベントに対する応答も示している。第１のタイプはブックマーク又はクリックイベントであり、それに応答して、アニメーションシーケンス中に１つのノードの位置が移動され、他の要素がその１つのノードの動きに合わせて移動する。第２のタイプは挿入/削除イベントであり、それに応答して、ある要素が収縮され別の要素が拡張されるアニメーションシーケンス中に、１つのノードは安定したままであるが、他の要素は収縮及び拡張に合わせて移動する。

【００６７】ブックマーク又はクリックイベントは、ユーザがメニュー又は他のブックマークの集合の中の項目を選択したとき、又はマウスダウンアップ（マウスボタンを押して放す）クリックで表現内の位置を選択したときに生じ得る。このタイプのイベントに応答して、クライアント２１２はノードＩＤ及びユニットディスク内の宛先位置を入手する。ブックマークイベントの場合は、ノードＩＤ及び宛先位置は以前に格納されており、メモリから検索できる。クリックイベントの場合は、クライアント２１２は、本明細書に参照として援用する米国特許第5,590,250号のコラム７１－７２に関係して記載されている“最も近いノードを探索”という関数（機能）とほぼ同じ方法で、選択位置から最も近いノードのノードＩＤを入手してもよく、宛先はユニットディスクの中心のようなデフォルト位置であってよい。

【００６８】処理ボックス３２０では、クライアント２１２はウォーカルーチン２２２をノードＩＤ及び宛先位置で呼出してもよい。ウォーカルーチン２２２は、ノードが前の位置から宛先位置へと移動する表現のシーケンスを表示するためのアニメーションループを実行することによって応答できる。処理ボックス３２０では、ウォーカルーチン２２２は、個々がノードＩＤ及びユニットディスク内の位置を含むノード/位置の対のシーケンスをセットアップすることによって開始する。位置は、米国特許第5,619,632号の図１２のボックス４７０、４７２、及び４８２に記載されているように、前の位置から宛先位置への総トランスレーション（移動路）を入手して、次に総トランスレーションのｎ番目のルートを入手して現在のトランスレーションと共に構成することを繰り返すことによって入手できる。ノード/位置の対の数は、アニメーション表示中に構造の要素を表わすフィーチャがオブジェクト不変性を維持しながら、前の位置から宛先位置への滑らかなアニメーションを確保するために、十分に大きくできる。ｎ番目ルートの方法の代わりに、前の位置から宛先位置へと双曲平面内で適切に選択された弧に沿って適切な数の点を選択することによって、位置を入手してもよい。弧は、不自然に見え得る直線と、ｎ番目ルートの方法でノードがとったであろう、滑らかに見えるには過度のアニメーションステップ数を必要とし得る弧との間を折衷するように選択されてもよい。点の数は、満足なアニメーションを確保するように選択されてもよい。

【００６９】次に、ウォーカルーチン２２２は、判断ボックス３２２で示されるように、シーケンス中のノード/位置の各対についてアニメーションループを反復する。処理ボックス３２４では、ウォーカルーチン２２２は、処理ボックス３１２に関係して上述したように、まず双曲平面にツリーの任意の保留中の編集をレイアウトしてもよい。次に、処理ボックス３２６では、ウォーカルーチン２２２は、処理ボックス３１４に関係して上述したように、ツリーをユニットディスクに、次のノード/位置の対からのノード及び位置を開始ノード及び開始位置として開始して、マッピングしてもよい。

【００７０】ツリーがマッピングされたら、処理ボックス３２８でマッピングされたツリーを表示バッファにペイントするためにペインタルーチン２２４を呼出すことができる。ペイント中、ペインタルーチン２２４は、処理ボックス３１６に関連して上述されたように、ノード生成の結果ツリー内で生じる新たな編集をマーキングできる。ペイントが完了したら、ツリーをペイントされたように表示するために表示バッファのスワッピングを実行して、グラフの表現を提供することができる。

【００７１】処理ボックス３２８でペインタルーチン２２４によって新たな編集がマーキングされると、次の反復中、処理ボックス３２４で、新たな編集がレイアウトされる。その結果、表現のアニメーション化されたシーケンスは、米国特許第5,629,632号のように静的ノードーリンク構造を示すのではなく、動的ノードーリンク構

造を示す。しかしながら、この編集は、基本的には、表現が前の位置から宛先位置へと遷移するときに表現の外周に沿って新たなノードを表わすフィーチャを追加するために働く。その結果、追加されたフィーチャは、他の要素を表わすフィーチャについてのオブジェクト不変性の知覚を妨害又は低下させない。

【００７２】挿入/削除イベントは、ユーザがノードの拡張又は収縮を要求したときに、又はグラフ又はツリーの何か他の変更を要求したときに生じ得る。挿入/削除イベントは、呼出しの形態でも受信されてもよく、従って、人の手による制御を並行させないグラフ又はツリーの自動変更のための機構を提供してもよい。更に、挿入/削除イベントは、全ての変更のアニメーションに対する呼出しを保留して得られた変更を示すデータを有する、２つ以上の個別の変更を要求する信号から生じてもよい。１つの挿入/削除イベントが、削除及び挿入の両方を含んでもよい。

【００７３】このタイプのイベントに応答して、クライアント２１２は、処理ボックス３３０で、要求されたグラフ又はツリーの変更が許容可能か否かを決定するために、まずメモリ２１４内のルーチンに対する適切な呼出しを行うことができる。

【００７４】要求されたグラフ又はツリーの変更が許容可能な場合は、それに従って、クライアント２１２は、必要に応じてメモリ２１４内のルーチンを呼出して、グラフ又はツリーを変更できる。変更を行うプロセスでは、変更によって挿入、削除、又は変化させられ得る、本明細書では“影響された要素”と呼ぶ各要素が、フラグの設定又は他の適切なデータの格納等によってマーキングされる。あるノードが拡張信号又は収縮信号によって選択された場合は、その親に割当てられた領域が変化し得るので、その親も影響されたノードである。他の挿入/削除イベントのほとんどについては、挿入又は削除されたノードだけが影響される。クライアント２１２は、あるノードが、変更のアニメーション表示中もそのノードの前の位置に保持される、安定ノードとして選択できる。

【００７５】多くの場合、前にマッピングに用いられた開始ノードを安定ノードとして選択することができ、そのノードを前の開始位置に保持することができる。他の場合には、クライアント２１２にとって、異なる安定ノードの選択が望ましいことがある。例えば、拡張されているノードを、現在の位置に保持される安定ノードとして選択してもよく、従ってそのノードが新たな開始位置となる。従って、クライアント２１２が異なる安定ノードを選択しない限り、通常は前の開始ノード及び開始位置が保たれる。しかし、前の開始ノードが削除されている場合は、他のノードがクライアント２１２によって変更されるデフォルト安定ノードとして選択されなければならない。

【００７６】削除が行われているとき、削除中のノード及びマッピング中のツリー内に削除後も残る最も近い祖先のノードＩＤでウォーカルーチン２２２を呼出すことができる。この祖先は、削除中のノードから現在の挿入/削除イベントによって削除されていない祖先に到達するまで上向きに巡回する(walk)ことによって、見つけることができる。

【００７７】この呼出しに応答して、ウォーカルーチン２２２は、削除中のノードが前の開始ノードか否かをテストできる。そうである場合は、識別された祖先を、開始ノードとして、削除中のノードと置換するために選択することができる。祖先が表示されている位置に最近マッピングされたものであり、それが使用可能な位置である場合は、その位置を開始位置として選択できる。祖先が最近マッピングされたものではない、又は、表示されない位置又は現在他の要素がマッピングされているために使用不可能な位置にマッピングされたものである場合は、開始位置をユニットディスクの中心にすることができる。

【００７８】処理ボックス３３０でも、クライアント２１２は安定ノードのＩＤ及び位置でウォーカルーチン２２２を呼出してもよい。ウォーカルーチン２２２は、まず削除されたノードが前の位置で収縮され、次に挿入されたノードが新たな位置で拡張され、その間ずっと、安定ノードは前の位置に保持されるという、表現のシーケンスを表示するアニメーションループを実行することによって応答できる。安定ノードが、最近マッピングされたものではない、又は表示されない位置又は使用不可能な位置にマッピングされたものであるので前の位置に保持できない場合は、前の開始ノードが前の開始位置にある状態で、削除されたノードが収縮された後に、安定ノードをその位置に移動することができ、その結果、収縮と拡張との間の突然の動きが生じる。ウォーカルーチン２２２は、収縮及び拡張中、影響された各ノードに割当てられた領域が変化するレートを支配する、重みのシーケンスをセットアップすることによって開始する。重みは、アニメーション表示中のオブジェクト不変性を保つために、十分に小さい増分で区分されている。

【００７９】次に、ウォーカルーチン２２２は、判断ボックス３３２で示されるように、シーケンス中の各重みについてアニメーションループを反復する。処理ボックス３３４では、ウォーカルーチン２２２は、反復の重みを用いて、まず双曲平面に影響されたノード及びツリーの任意の保留中の編集をレイアウトしてもよい。次に、処理ボックス３３６では、ウォーカルーチン２２２は、処理ボックス３１４に関係して上述されたように、ツリーをユニットディスクに、安定ノード及び位置で開始して、マッピングしてもよい。

【００８０】ツリーがマッピングされたら、処理ボックス３３８で、マッピングされたツリーを表示バッファに

ペイントするために、ペインタルーチン２２４を呼出すことができる。ペイント中、ペインタルーチン２２４は、処理ボックス３１６及び３２８に関係して上述されたように、ノード生成の結果ツリー内で生じる新たな編集をマーキングできる。ペイントが完了したら、ツリーをペイントされたように表示するために表示バッファのスワッピングを実行して、グラフの表現を提供することができる。

【００８１】保留中の編集が存在してもしなくても、判断ボックス３３２で開始するアニメーションループの一連の反復は、削除及び/又は挿入による動的ノード－リンク構造の表現を生成する。更に、影響された要素は、削除及び挿入の前の位置から新たな位置へと移動する。この技術は、これらの動きの間のオブジェクト不変性を生み出すために、うまく実現された。

【００８２】処理ボックス３１６で表現が供給された後、又は判断ボックス３２２又は３３２でアニメーションシーケンスが完了した後、図６の“Ａ”と表示された円によって示されるように、処理ボックス３０２で別のイベントを受信できる。

【００８３】図７は、図６の処理ボックス３００でレイアウトが最初にどのように実行できるかを示している。図８は、処理ボックス３１２、３２４、及び３３４で、変更されたノード－リンク構造のレイアウトがどのように実行できるかを示している。

【００８４】処理ボックス３５０で示されるように、ウォーカルーチン２２２は、ルートノードＩＤを入手して有向グラフデータ構造２３２内のルートノードに関係するデータにアクセスするために用いることによって、最初のレイアウトを開始する。処理ボックス３５２では、ウォーカルーチン２２２は、角度幅で数学ルーチン２２６に対する呼出しを行うことによってルートノードをレイアウトする。この角度は、望ましい結果を生成する任意の適切な角度でよい。角度2π及びπ/2がうまく用いられ、2πを用いた場合は中心レイアウトスタイルに適しており、π/2を用いた場合は上、下、右、又は左レイアウトスタイルに適していた。望ましい結果を得るためにこの角度を修正するために、インタフェースが提供されてもよい。

【００８５】応答では、数学ルーチン２２６はルートノードを、ユニットサークルの原点、即ち座標（0,0）に、上向きのオリエンテーションで座標（0,1）に、及び角度幅の半分の角度でレイアウトする。次に、処理ボックス３５４で、ウォーカルーチン２２２はルートノードＩＤをキューの前にプッシュする。

【００８６】図７の残りでは、判断ボックス３６０で示されるように、ウォーカルーチン２２２は、有向グラフデータ構造２３２によって定義されるツリーの１組の要素を、キューが空になるまで、繰返し走査する。各反復は、キューの後からノードＩＤを入手し、それを有向グラフデータ構造２３２内の識別されたノードに関係するデータにアクセスするために用いることによって開始する。

【００８７】各反復で、判断ボックス３７０でのテストは、ノードがこの走査で既に巡回されたか否かを決定する。巡回されていない場合は、ウォーカルーチン２２２は処理ボックス３７２で巡回済ノードをマーキングし、複数の子のノードＩＤを入手し、複数の子のうちの非拡張リーフ、即ち拡張された入リンクを持たないリーフノードを識別し、以下に説明するように可視ツリー内に存在する子の数を入手し、同じく以下に説明するように、可視ツリー内の複数の子についての角度及び半径のアレイを入手するために、可視ツリー内に存在するノードの子の数Nで数学ルーチン２２６を呼出す。

【００８８】処理ボックス３７２での処理については、近隣関係データの１タイプである可視ツリー内の子の数Nは、２通りの方法のいずれかで計算できる。即ち、現在の走査がノード追加ステップのシーケンスの一部である場合は、Nは走査前の子の数と追加された子の数との合計に等しい。現在の走査がノード除去ステップのシーケンスの一部である場合は、Nは単に走査前の子の数に等しい。

【００８９】処理ボックス３７２では、角度及び半径のアレイは様々な方法で入手できる。１つの成功した実施例では、各半径は0.7の値に設定され、一方、各角度は((N*π)/18)とπとのより小さい方に設定される。従って、N<18については、ノードの角度は可視ツリー内に存在するそのノードの子の数に左右される。

【００９０】次に、処理ボックス３７４で、ウォーカルーチン２２２は子をレイアウトするために数学ルーチン２２６を呼出す。

【００９１】処理ボックス３７４を実施する際、レイアウトの２つの一般原則が適用される。第１に、ノード間の離間及び角度は、ツリー内の近隣要素に関する情報、即ち近隣関係データだけに基づいて決定される。第２に、各ノードについて入手されたレイアウト情報は、あるノード及びその全ての子の位置がデータ構造内の小さな変化で移動できるような方法で、ノード対その親の相対位置を示す。

【００９２】従うことのできる一般戦略は、処理ボックス３７２からの子の半径及び角度で開始し、子が必要とする近似距離を入手し、近似距離を用いて親からの距離を入手し、親からの距離を用いて子についてのより正確な距離を入手し、次に、随意的により正確な距離を用いて更により正確な距離を入手する、等である。

【００９３】一般戦略に従って、子が処理ボックス３７２からの半径R及び角度Θを有する場合は、近似距離D1及びD2をそれぞれsinh(R)及びtan(Θ/4)として計算できる。Nがレイアウト実行中の子の数であり、合計がi=1からN-1まで実行される場合に、それぞれの近接した子の

対が、それぞれのD1及びD2の合計のより大きい距離、即ちDT=Σ(max(D1(i)+D1(i+1),D2(i)+D2(i+1))+max(D1(1),D2(1))+max(D1(N),D2(N))によって分けられているとき、D1及びD2を用いて、全ての子についての総距離DTを得ることができる。その親が使用可能な角度ωを有する場合、親からの距離DPはasinh(DT/ω)として計算できる。次に、子を、親を中心とする半径DPの円の円周に沿って、子の分割に比例する子の間の角度で配置できる。

【００９４】次に、DPを用いて、子についてのより正確な距離を以下のように得ることができる。

D1'=sinh(DP)asin(sinh(R)/sinh(DP))、

D2'=2sinh(DP)atan(tan(Θ/4)/e^{DP})

次に、D1'及びD2'を用いて、上述の方法等のように、所望のレベルの正確さに到達するまで、親までのより正確な距離DP'を得ることができる。その時点で、それぞれの子のオリエンテーションを親のオリエンテーションからの角度のずれとして計算できる。

【００９５】上述の距離は双曲平面内の実際の計量で表わされていることに注目されたい。双曲平面内の距離Dは、ユニットサークル内の原点で開始してtanh(D/2)の距離を進むベクトルに対応する。

【００９６】処理ボックス３７２に関係して上述したように、一般戦略はこのようにして、ノード、その親、及び可視ツリー内の同胞の子に関する情報を含む同胞に関する近隣関係情報だけに基づくレイアウト情報を入手する。一般戦略は、子からその親までの距離と、子とその親とのオリエンテーションの違いを表わす角度とを示す、レイアウト情報を入手する。

【００９７】一般戦略は、全ての子について２つの反復ループを遂行するが、一般戦略に関係して上述した方法でより正確な距離を得る試みをせずに得られる最初の距離を用いるソフトウェアとして実施された。第１のループは、隣接する子の間の分離及び各子についての“スライスサイズ”を入手して一時的に保存し、分離の合計も入手する。次に、この情報を用いて親までの距離を得る。次に、第２ループは、各子の相対的なオリエンテーション及び領域を入手して保存する。

【００９８】このソフトウェアとしての実施例では、子が処理ボックス３７２からの半径R及び角度Θを有する場合は、一般戦略に関係して上述したように距離D1及びD2が計算される。各子についてのD1及びD2を前の子についてのD1及びD2に加え、S1及びS2を得る。分離の合計STは、最初と最後の子についてを除き、子のS1及びS2の最大値に増加され、最初と最後の子については、STはその子のD1及びD2の最大値に増加される。

【００９９】子のS1がその子のS2より大きい場合は、S1が子の分離として保存されると共にまずD1が子のスライスサイズとして保存され、２番目以降の子については、前の子のスライスサイズがその子の前のスライスサイズとその子のS1との最小値となるように調整される。逆に、子のS1がその子のS2より大きくない場合は、S2が子の分離として保存されると共にまずD2が子のスライスのサイズとして保存され、２番目以降の子については、前の子のスライスサイズがその子の前のスライスサイズとその子のS2との最小値となるように調整される。しかしながら、最後の子のスライスサイズは、その子の前のスライスサイズとその子のD1及びD2の最大値との最小値となるように調整され、第１の反復ループが完了する。

【０１００】次に、親の角度ωを用いて、親からの距離DPをtanh(asinh(ST/2ω)/2)又は0.5の大きい方として計算することができる。DPは親ノードに関連するデータの一部として保存される。

【０１０１】第２の反復ループは、各子について、Sが子についての保存された分離であるとき、角度(S/ST)2ωを計算することによって開始する。角度(S/ST)2ωは、-2ωで開始したランニングトータルに加えられる。ランニングトータルは子に関連する他のデータと共に保存される。

【０１０２】次に、数学ルーチン２２６は、米国特許第5,590,250号のコラム６７及び６８の“内角(inside-angle)”関数に類似の関数を呼出すことによって、子についての新たな角度を計算できる。この、本明細書では“インサイドアングル(InsideAngle)”と呼ぶ関数は、楔の中へと移動された距離(“dist”)及び楔の半分の角度で開始する。インサイドアングルは作動可能角度として、開始角度と、εが0.0001のような非常に小さい値を有し得るときの（π－ε）との、小さい方をとり、逆タンジェントの計算における問題を回避する。インサイドアングルは、ユニットサークル上の座標(dist,0)にある点を原点へと移動する変形を入手する。次に、インサイドアングルは、ユニットサークルの円周と、原点から発する、水平位置との角度が作動可能角度である放射線との交点にある点の複合座標に、この変換を適用する。インサイドアングルは、結果の角度として、原点から変換された点を通る放射線の水平位置からの角度を戻す。

【０１０３】子の角度を得るために、インサイドアングルは、距離DP及び第１反復からの子のスライスサイズを２ω/STで乗算することによって計算された角度で呼出される。インサイドアングルによって戻された角度はπ/2と比較され、子の角度はこの２つのうちの小さい方である。

【０１０４】子の新たな角度を保存する前に、数学ルーチン２２６は子の前の角度を保存する。古い角度と新しい角度との差の絶対値が最小値を越える場合は、数学ルーチン２２６は、以下に述べるように、レイアウトが継続すべきであることを示すデータも保存する。

【０１０５】最後に、第２反復ループは、米国特許第5,590,250号のコラム６７及び６８の“空間使用可能(room-available)”関数（機能）と類似の関数を呼出すことによって、子の領域又はサイドスペースを得る。本明細

書では“ルームアベイラブル(RoomAvailable)”と呼ぶこの関数は、楔の中へと移動された距離D及び楔の半分の角度Φで開始する。ルームアベイラブルは、まず比$(1-D^2)/2D$を得ることによって、次に初期距離Sを得るために比をsinΦで割ることによって計算される、楔のエッジまでの距離を戻す。次に、ルームアベイラブルは距離$((S^2-1)^{1/2}-S)$を戻す。子の領域を得るために、ルームアベイラブルは上述のインサイドアングルの呼出しに用いられたのと同じ距離及び角度で呼出される。ルームアベイラブルによって戻された距離は子の領域の尺度として保存される。

【０１０６】上述のソフトウェアとしての実施例は付加的なデータを保存できるにも関わらず、ソフトウェアとしての実施例は、本明細書に述べるように、レイアウト及びマッピングを実行できるためには、各ノードについて２項目のデータだけ格納される必要があるという発見に基づいている。一方の項目は、距離即ち双曲平面内のノードからその子ノードまでの位置変位を示す。他方の項目は、そのノードの親までの入リンク及びその親からそのノードへの出リンクの拡張の間の双曲平面内の角度変位である。これらの２項目のデータ、又はそれらにアクセスするために用いることができるハンドルは、有向グラフデータ構造内のリンクのデータ項目に含まれることができる。

【０１０７】判断ボックス３８０のテストは、ノードの次の世代までレイアウトを継続するか否かを決定するために適切な基準を適用する。処理ボックス３７４に関係して上述されたように、基準は、任意の子ノードの角度が0.00001のような小さな角度の違い以上の変更をされているか否かであり得る。そうである場合は、レイアウトを継続すべきである。

【０１０８】処理ボックス３８２では、ウォーカルーチン２２２は、拡張されている又はリーフではない各子ノードのIDを、キューの前にプッシュする。他の子ノードは、レイアウトされる子を持たないので、処理ボックス３８２でマーキング巡回されてよい。処理ボックス３８２が完了すると、又は判断ボックス３８０のテストが継続しないと決定すると、又は判断ボックス３７０のテストがノードがすでに巡回されたと決定すると、判断ボックス３６０に戻る前に処理ボックス３８４でキューの後のノードが取り出される。

【０１０９】図８は、図６の処理ボックス３１２、３２４、及び３３４で、どのように変更されたノードーリンク構造のレイアウトが実行できるかを示している。それぞれの場合に、レイアウトは、処理ボックス４００で示されるように、レイアウト及びマッピングへと導く呼出しに応答して開始する。しかしながら、判断ボックス４０２のブランチによって示されるように、レイアウトが実行される方法はノードーリンク構造になされた変更のタイプによる。

【０１１０】変更が、オリエンテーションイベントに応答したルートノードのオリエンテーションの変更である場合は、処理ボックス４０４で、ウォーカルーチン２２２は数学ルーチン２２６を呼出し、マッピング及びペイントの前にルートノードを新たなオリエンテーションにレイアウトすることができる。ルートノードは、新たなオリエンテーション以外では、図６の処理ボックス３５２に関係して上述されたようにレイアウトできる。次に新たなオリエンテーションがマッピングに用いられ、表現のオリエンテーションを変える。

【０１１１】変更が、ストレッチイベント、ドラッグイベント、ブックマークイベント、又は編集が保留中の場合のクリックイベントに対する応答で生じ得るように非アニメーション編集である場合は、ウォーカルーチン２２２は、処理ボックス４１０で、まず除去編集のリストをセットアップし、次に処理ボックス４１２で、マッピング及びペイント前に除去編集をレイアウトする。次に処理ボックス４１４で、ウォーカルーチン２２２は追加編集のリストをセットアップし、次に処理ボックス４１６で、マッピング及びペイント前に追加編集をレイアウトする。

【０１１２】この実施では、編集のリストは、グラファルーチン２２０及びペインタルーチン２２４を含むメモリ２１４内の種々のルーチンによって維持される編集ソースリストに基づいてセットアップされる。現在の実施は、拡張されたリンクによって定義されるツリーにも関係する。本明細書で“コラプストリンクス(CollapsedLinks)”及び“エキスパンデッドリンクス(ExpandedLinks)”と呼ぶ１対の編集ソースリストは、それぞれ収縮要求及び拡張要求によって選択されたリンクについての編集を含み、従って、図６の処理ボックス３３０でセットアップできる。本明細書で“リムーブドリンクス(RemovedLinks)”及び“アディッドリンクス(AddedLinks)”と呼ぶ他の対は、それぞれ削除及び挿入されたリンクについての編集を含む。編集ソースリストのコピーは、異なる目的で複数存在していてもよい。

【０１１３】処理ボックス４１０でセットアップされた除去編集のリストはリムーブドリンクスに基づいており、一方、処理ボックス４１４でセットアップされた追加編集のリストはアディッドリンクスに基づいている。処理ボックス４１０又は４１４でリストをセットアップする際に、ウォーカルーチン２２２は適切な編集ソースリスト内の各編集にアクセスし、その編集用いてセットアップ中のリストについての適切なエントリを入手する。それぞれの場合に、編集ソースリスト内の編集を用いて、編集のリンクの子ノードのノードＩＤ及び実行中の編集のタイプを示す編集識別子を入手する。

【０１１４】その子ノードの親は、既にリスト上に存在するのではない限り、影響されたノードのリストの後に加えられる。親は、その子ノードの拡張された親ノード

か、又は、その子ノードの親ノードが現在どれも拡張されていない場合は、その子ノードの最初の親ノードである。編集ソースリストからの編集が親を持たない子ノードに関係する場合は、その編集はそのルートに関係しなければならず、従ってその場合は、そのルートノードが影響されたノードのリストの後に置かれる。

【０１１５】ソース編集リストからのリンクについての反復の最後に、次の反復に用いられる適切な編集ソースリスト内の次の編集にアクセスする前に、そのリンクの子ノードも子ノードリストに追加される。このように、影響されたノード及び子ノードのリストを完成するために全ての編集が処理されるまで、編集ソースリストの各編集について反復が行われる。

【０１１６】次に、処理ボックス４１２又は４１６では、リストを用いて編集がレイアウトされ、影響されたノードのリスト内の各ノードについて図７の処理ボックス３５４から３８２のシーケンスに類似のシーケンスを辿り、ソフトウェアのノードをリストから、ルートノードよりもキューの前にプッシュし、処理ボックス３７２で、次のように幾つかの変更を行う。処理ボックス４１２及び４１６のレイアウトは、どの子が非拡張リーフであるかの識別に加えて、各子が子ノードリストに存在するか否かを決定する。そうである場合は、レイアウトはその子についての角度及び半径を重みで乗算する。処理ボックス４１２では、重みが０なので、処理ボックス３７４でその子はほぼその前の位置に角度及び半径が０でレイアウトされ、従って消滅する。処理ボックス４１６では、重みが１なので、処理ボックス３７４でその子は新たな位置に全角度及び全半径でレイアウトされる。

【０１１７】処理ボックス４１０から４１６の操作もアニメーションシーケンス内で実施されてもよく、その場合は、除去編集はアニメーションシーケンスの最初の部分で処理されてもよく、追加編集はシーケンスの後続部分で処理されてもよい。一方、図６に関係して上述したように主にペイント中のノード生成から非アニメーション編集が生じる場合は、その編集は追加編集だけであってよく、アニメーションシーケンスの各ステップで全ての現在保留中の編集が処理されてもよい。

【０１１８】要素を収縮又は拡張する要求のように挿入/削除イベントに応答して、又はそのようなイベントによって要求された多数の個別の変更をアニメーション化する要求に応答して生じ得るように、変更がアニメーション化編集である場合は、ウォーカルーチン２２２は、処理ボックス４２０で、まずソース編集リストに基づいて除去及び追加される要素の数を入手する。除去される数は、コラプストリンクス及びリムーブドリンクス内の要素の数を加算することによって得ることができ、一方、追加される数は、エキスパンデッドリンクス及びアディッドリンクス内の要素の数を加算することによって得ることができる。次に、処理ボックス４２２で、ウォーカルーチン２２２は、除去ステップと追加ステップとの間に使用可能なアニメーションステップを割付け、幾分図８の処理ボックス４１０及び４１４のように、除去編集及び追加編集のリストもセットアップする。アニメーションステップの単純な割付けは、半分の除去ステップと半分の追加ステップであるが、除去される要素が無い場合は全てのステップが追加ステップとなり得、追加される要素が無い場合はその逆である。

【０１１９】処理ボックス４２２での除去編集及び追加編集リストのセットアップにおいて、ウォーカルーチン２２２は、収縮又は拡張されたノードが存在しない限り、処理ボックス４１０及び４１４に関係して上述したように行うことができる。収縮又は拡張されたノードの場合は、ノード自体が、その親に加えて、影響されたノードリストの後にプッシュされ、次に、そのノード自体ではなくそのノードの子が、子ノードリストに加えられる。言い換えれば、収縮又は拡張は、１つのノードだけに影響する他の操作とは異なり、２世代のノードに影響すると考えることができる。ウォーカルーチン２２２は、影響されたノード及び子ノードの２対のリストをセットアップし、１対を除去編集に、もう１対を追加編集に用いる。

【０１２０】次に、判断ボックス４３０で開始するループで、ノードを除去するアニメーションステップが実行され、処理ボックス４３２で、重みを入手し、重みを用いて除去編集をレイアウトし、アニメーションフレームをマッピング及びペイントする。同様に、次に判断ボックス４４０で開始するループでノードを追加するアニメーションステップが実行され、処理ボックス４４２で、重みを入手し、重みを用いて追加編集をレイアウトし、アニメーションフレームをマッピング及びペイントする。ノードを追加する前にノードを除去することによって、単一フレーム内で同一ノードが２箇所に出現する状況が防止される。ノードを除去した後に１つの最終ステップを重み０で実行し、ノードを追加した後にもう１つの最終ステップを重み１で実行することによって、この技術は最終重みがそれぞれ０又は１であることを確実にできる。

【０１２１】処理ボックス４３２で、除去アニメーションステップ数から現在の除去アニメーションステップ番号を引き、次にその差を除去アニメーションステップ数で割ることによって、一連の除去アニメーションステップの間に重みが１から０になるように重みを入手できる。同様に、処理ボックス４４２で、現在の追加アニメーションステップ番号に１を加え、次にその合計を追加アニメーションステップ数で割ることによって、一連の追加アニメーションステップの間に重みが０から１になるように重みを入手できる。

【０１２２】アニメーションステップの総数は、アニメーション速度と共に、アニメーション表示中のオブジェ

クト不変性の知覚への影響を助ける。上述の重みを得る方法の説明から理解できるように、アニメーションステップの総数は、除去又は追加された要素の領域が変化するレートを決定し、従って、他の要素が除去又は追加された要素の領域に関係して移動しなければならないレートを間接的に決定する。十分なアニメーション速度が維持されることを前提とすれば、除去ステップと追加ステップとの間に適切に割付けられたアニメーションステップ数が多い方が、オブジェクト不変性を生み出すのにより適している。

【０１２３】図８の技術は、適切な速度の適切な数のアニメーションステップで実行されると、幾分閉じられたり広げられたりする扇のような、収縮及び拡張する１組のノードの知覚をうまく作り出した。ノードに割当てられた半径及び角度を調整することによって、削除されたノードがその親に引き込まれる又は無限へと絞り出される、又は挿入されたノードがその親から生え出る又は無限から引っぱり込まれるような、異なる知覚を得ることができる。１つの子のグループから１つの子だけ削除される場合には、その子が無限へと絞り出されるように見え、収縮のように１グループとして全ての子が削除される場合には、全ての子が親に引き込まれるように見えることが可能である。同様に、１つのグループに１つの子が追加される場合には、その子が無限から引っぱり込まれるように見え、拡張のように１グループとして全ての子が挿入される場合には、全ての子が親から生え出るように見えることが可能である。更に、孫が安定して見え、子だけが移動して見えるように調整されたレートで、子が親に引き込まれている間に孫を無限へと絞り出すことができる。

【０１２４】上述のように、編集リスト内で示される２つ以上の個別の変更を並行してアニメーション化することが可能である。更に、クライアント２１２が、インタフェースを介して２つ以上の個別の変更がアニメーションなしで行われるように要求できるように、インタフェースを備えることができる。言い換えると、編集リストは、編集の完遂を要求するコマンドが受信されるまで形成され続けることが可能である。コマンドは、編集が、上述のようにアニメーションで、又はリフレッシュ動作のようにアニメーションなしで完遂されることを要求できる。編集が完遂されたら、編集リストを消去できる。

【０１２５】アニメーションを容易にするために、付加的な拡張/収縮情報を維持できる。例えば、追加のフラグが、各表現についてどの要素が走査されるべきかに関する急速な決定を可能にすることによって、前の構造の表現と変更された構造の表現との迅速な交換を可能にし得る。

【０１２６】図９及び１０は、現在のソフトウェアとしての実施で表示されるようなノード－リンク構造の表現の例示的なシーケンスを示しており、図４から８に関係して上述されたシーケンスと実質的に同一である。図示されている表現は組織図を示しており、米国特許第5,619,632号の図１７から２１の表現にやや似ている。

【０１２７】図９は、ノードフィーチャを収縮する要求から生じ得る表現のシーケンスを示しており、一方、図１０は、ノードフィーチャを拡張する要求から生じ得る表現のシーケンスを示している。また、図９から図１０を通して連結されたシーケンスは、以下に説明するように、前に拡張されたノードフィーチャのクローンを拡張する要求から生じ得る。

【０１２８】図９の表現５００、５０２、及び５０４は、組織図内の個人を表わすノードフィーチャを含む。各ノードフィーチャは、最大サイズで表示されると、人物の顔のグラフィック表現及びその人物の名前又はニックネームを有する矩形のタイトルブロックを含む。子孫を有するノードの矩形のタイトルブロックの下右隅には、それぞれがノードフィーチャの拡張又は収縮を要求するための小さな“＋”又は“－”があってもよい。これらの小さなシンボルは、選択された後、例えば第１アニメーションフレーム内のように、適切な時に遷移を行うことができる。ノードフィーチャが最大サイズよりも小さく表示されると、グラフィック表現は縮小されたサイズを有することができ、タイトルブロックは短かくすることができる。ノードフィーチャは、表現の周辺部に近付くと小さな点になることができ、次いで見えなくなることができる。

【０１２９】ノードフィーチャ５１０は表現５００、５０２、及び５０４のそれぞれに存在しており、表現５００では拡張されて示され、子が周辺部近くの小さな点で示されている。図９のシーケンスは、例えば、ノードフィーチャ５１０を収縮する要求から生じ得る。この要求に応答して、ノードフィーチャ５１０によって表わされるノードの子及び孫がノード－リンク構造から除去され、表現５００から子及び孫が不在の表現への遷移がアニメーション化される。表現５０２及び５０４は、そのようなアニメーション化シーケンスで表示され得る２つの中間表現である。

【０１３０】表現５０２では、ノードフィーチャ５１０の子がノードフィーチャ５１０へと引かれ始めたところである。しかしながら、ノードフィーチャ５１０の孫は、子がノードフィーチャ５１０に向かって引かれるレートを相殺するレートで子から広がっているので、所定位置に留まる。

【０１３１】表現５０４では、子はノードフィーチャ５１０に向かって更に近くに引かれており、一方、孫は所定位置に留まり続けている。更に、子は表現の周辺部から離れたので、各子の矩形のタイトルブロックが表示できる十分な領域を有する。タイトルブロックはペイントされた順に基づいて重なり合い、１組の扇状に広げられたカード又はシートの知覚を生み出している。

【０１３２】ノードフィーチャ５１０の子が辿るパスは、全ての子がノードフィーチャ５１０に収束するので、平行ではない。子がノードフィーチャ５１０に近付くと、ノードフィーチャ５１０の領域が減少し、人物の顔のグラフィック表現のサイズが一時的に縮小される。

【０１３３】表現５０４には、潜在的にあと幾つかの中間表現の後、ノードフィーチャ５１０の子又は孫が全く表示されない最終表現が続く。この様子は、ノードフィーチャ５１０が最終表現内で見えるように見える図１０の表現５４０から理解できる。

【０１３４】図１０では、表現５４０、５４２、及び５４４のそれぞれが、図９の表現５０４では収縮されて示されているが図１０の表現５４４では拡張されて示されているノードフィーチャ５５０を含んでおり、子が周辺部近くの小さな点で示されている。図１０のシーケンスは、例えば、ノードフィーチャ５５０が表現５０４で見えるように見えるときにノードフィーチャ５５０を拡張する要求から生じ得る。この要求に応答して、ノードフィーチャ５５０によって表わされるノードの子及び孫がノード－リンク構造に追加され、表現５０４から子及び孫が存在する表現への遷移がアニメーション化される。表現５４０及び５４２は、そのようなアニメーション化シーケンスで表示され得る２つの中間表現であり、表現５４４は最終表現である。

【０１３５】表現５４０では、ノードフィーチャ５５０の子がノードフィーチャ５５０から広がり始めたところである。子はまだ表現の周辺部からある距離だけ離れているので、各子の矩形のタイトルブロックが表示できる十分な領域を有する。図９の表現５０４のように、タイトルブロックは重なり合い、１組の扇状に広げられたカード又はシートの知覚を生み出している。しかしながら、ノードフィーチャ５５０の孫は、子がノードフィーチャ５５０から広がるレートを相殺するレートで無限から子に向かって引かれるので、既に所定位置にある。

【０１３６】表現５４２では、子はノードフィーチャ５５０から更に離れて広がっており、一方、孫は所定位置に留まり続けている。更に、子は表現の周辺部により近付いたので、タイトルブロックを表示するための十分な領域を持たず、小さな点となっている。

【０１３７】最後に表現５４４では、ノードフィーチャ５５０の子は表現の周辺部に近い位置に到達している。周辺部に近いことによって、孫は見えない。

【０１３８】ノードフィーチャ５５０の子が辿るパスは、全ての子がノードフィーチャ５５０から拡散するので、平行ではない。表現５４０で最初に子が出現するときには、子はノードフィーチャ５５０に近接しているので、ノードフィーチャ５５０の領域は減少し、その結果、人物の顔のグラフィック表現のサイズが一時的に縮小される。子がノードフィーチャ５５０から分散するにつれて、ノードフィーチャ５５０の領域は増大する。

【０１３９】上述したように、図９及び図１０のシーケンスを連結して、まずノードフィーチャ５１０が収縮され次にノードフィーチャ５５０が拡張される、単一のシーケンスを生成してもよい。これは、例えば、ノードフィーチャ５１０及び５５０が共にグラフ内の同一ノードを表わすのだが、そのノードが２つの入リンクを有するためにそのノードのクローンが生成され、図９及び１０で表わされるツリー内で２回発生する場合に生じ得る。ユーザが一方のクローンを表わすノードフィーチャの拡張を要求すると、その応答は前に拡張された他方のクローンを表わすノードフィーチャの収縮を含むことができる。実際には、そのブランチがノード５１０からノード５５０へと移動される。２つのノードフィーチャ内の“＋”及び“－”のシンボルは適切に変更される。

【０１４０】上述の実施に類似の実施は、ＩＢＭ互換ＰＣのプロセッサ上でうまく実行されたが、実施は任意の適切なプロセッサを有する他の装置で実行されてもよい。

【０１４１】上述の実施に類似の実施は、32ビットのウインドウズ(Windows)環境でC++言語を用いてうまく実行されたが、非オブジェクト指向環境を含む他のプログラム言語及び環境を用いてもよく、また、リスプ(Lisp)、ユニックス(Unix)環境、ANSI C、及びパスカル(Pascal)等のような他のプラットフォームを用いてもよい。

【０１４２】上述の実施に類似の実施は、ＸＭＬ準拠フォーマット及びある実験的フォーマットで表示されるノード－リンクデータを用いてうまく実行されたが、本発明は、静的又は動的いずれかの、メモリ内又はネットワークを介してのように任意の適切な方法でアクセス可能な、任意の適切なタイプのノード－リンクデータを用いて実行されてもよい。

【０１４３】上述の実施に類似の実施は、ナビゲーション信号に応答してグラフの１つの表現又はアニメーション化された一連の表現を準備及び表示する各反復を用いて実施されたが、本発明は他のタイプの信号又は呼出しによって呼出される他のタイプの反復を用いて実施されてもよい。

【０１４４】上述の実施に類似の実施は、キーボード及びマウスから受信され、且つノード－リンク構造の表示された表現又はアニメーション化された一連の表現に関係するナビゲーション信号を用いてうまく実行された。しかしながら、本発明は、ナビゲーション信号を用いて又は用いずに実施されてもよい。例えば、あるノードの子の異なるソーティングに応答して、又はある構造の要素への異なるフィルタの適用に応答して、要素を動き回らせてもよい。また、本発明は、任意の適切なタイプの拡張及び収縮信号、又は外部の照会、示されたノード又はリンクの下に拡張を要求するメニューエントリーのような項目の選択、又は現在の焦点の下に拡張を要求するメニューエントリーのような項目の選択から生じた信号

を含む他のナビゲーション信号を用いて実施されてもよい。ナビゲーション信号は、代わりに、ビデオゲーム又はバーチャルリアリティ環境によって生成される空間のような現実には無い空間又はディスプレイ以外の表示空間に関係してもよく、また、ナビゲーション信号は、代わりに、他の種類の位置決め装置、及び英数字又は声、身振りのような言語的入力、又は他の様式のユーザ入力を受信するための他の種類の装置を含む、任意の適切なユーザ入力装置によって生成されてもよい。本発明は、任意の適切なアニメーション技術を用いて実施されてよい。

【０１４５】上述の実施は、ノードが円、矩形、及びアイコン状の画像等のフィーチャによって表わされ、リンクがノードフィーチャを結ぶ線によって表わされるノードーリンク構造の表現を表示する。しかしながら、本発明は、少なくとも本発明の幾つかの態様について、インデントされた項目の子を下に有する項目の階層リストを含む表現における実施を含む、ノードーリンク構造を表現する任意の他の適切な方法を用いて実施されてもよい。

【０１４６】上述の実施は、時間及び空間の十分に小さいインターバルで類似のフィーチャを表示することによって、ステップのシーケンスを通して、共有ノード及びリンクを表わすフィーチャについてのオブジェクト不変性を維持するが、本発明はオブジェクト不変性を維持するための他の技術を用いて、共有ノードだけ又は共有リンクだけを表わすフィーチャについてのオブジェクト不変性を有して実施されてもよい。例えば、移動要素のオブジェクト不変性の提供を助けるために、位置間でのぶれ、位置間の輪郭のシーケンス、又は位置間の移動を示す他の標識のような他のキューを用いてもよい。

【０１４７】上述の実施では、削除又は挿入された要素を表わすフィーチャのための領域は、近隣要素の領域が増大又は減少する間に、それぞれ減少又は増大する。しかしながら、本発明は、周辺部から入る又は周辺部へと出る移動フィーチャ、削除及び挿入を示す色の変更、フェードイン又はフェードアウト、又はマイクロソフト社(Microsoft)のパワーポイント(PowerPoint)で使用可能な技術を含む任意の他の従来のアニメーション技術のような、削除及び挿入された要素を表わすフィーチャを変更する他の技術を用いて実施されてもよい。更に、本発明は、近隣要素の領域以外の領域を増大又は減少させる技術を用いて実施されてもよい。更に、削除はアニメーションなしで、アニメーション化され得る近隣要素の移動を補うアニメーションの始めに、即座に生じてもよく、同様に、挿入もアニメーションなしで、近隣要素の移動を補うアニメーションの終わりに、即座に生じてもよい。

【０１４８】上述の実施では、１組の要素が削除され、次に１組の要素が挿入され、要素の移動は、まず要素を

前の位置から削除し、次に要素を新たな位置に挿入することによって完遂される。しかしながら、本発明は、並行して要素を削除及び挿入する技術を用いて実施されてもよく、更に、本発明は、要素を削除して次に挿入する以外の方法で移動する技術を用いて実施されてもよい。

【０１４９】上述の実施は、ある特定の方法でレイアウトデータを得るが、本発明は、そのようなレイアウトを用いて又は用いずに実施されてもよく、又はノードーリンク構造全体を各表現について個別にレイアウトすることによる、又はノードーリンク構造を他の方法でレイアウトすることによるような、他の方法でレイアウトデータを得ることによって実施されてもよい。

【０１５０】上述の実施では、ノードーリンク構造はユニットディスクにマッピングされ、次にある特定の方法でペイントされるが、本発明は、マッピングを用いて又は用いずに実施されてもよく、又は、ノードーリンク構造を３次元レンダリング空間及び表示空間を含む任意の他の適切なレンダリング空間にマッピングして任意の他の適切な表示空間に表示することを含む、任意の他の適切な方法でノードーリンク構造をマッピング及び表示するように実施されてもよい。

【０１５１】上述の実施はツリーの表現の表示に適している。本発明は、一般的なグラフのような、他のタイプのノードーリンク構造の表現を表示するのに用いられてもよい。

【０１５２】上述の実施は、ある特定の方法で実行されるメモリ管理を用いて、グラフ内のツリーを定義するためのリンクの拡張フラグを含むノードーリンクデータを用いるが、本発明は、任意の他の適切な方法で定義され、任意の適切な方法でメモリにロードされるノードーリンク構造を用いて実施されてもよい。

【０１５３】上述の実施は、循環有向グラフを含む有向グラフを扱うことができるが、本発明は、他のタイプのリンクを有向グラフの適切な組合せに変換することによって、又は別様ではグラフの構造をツリーにマッピングするためのプロトコルを供給することによって、他のタイプのグラフ用に実施されてもよい。例えば、２つのノード間の無向リンクが、同じノードの間の１対の有向リンクに変換されてもよく、又は適切な基準に基づいて方向を割り当てられてもよい。一般的に、全ての無向リンクが１対の有向リンクへと変換されている表現は、結果的に有向リンクの各対が循環するので、視覚的に混乱させる傾向があるが、別の方法で循環を表示することによってこの混乱は克服されるかもしれない。

【０１５４】上述の実施では、処理は多くの場合に変更されてもよい順位で行われる。例えば、図６では、幅優先巡回よりも深さ優先巡回が実行されてもよい。

【０１５５】同じく、上述の実施では、幾つかのソフトウェアの部分が、グラファ、ウォーカ、ペインタ、及び数学ルーチン、並びにクライアントのように区別される

が、本発明は、ハードウェア及びソフトウェアの他の組合せ並びに任意の適切な方法で構成されたソフトウェアで実施されてもよい。

【０１５６】本発明は、ノード－リンク構造の対話式ブラウザの提供に適用された。本発明は、ノード－リンク構造が視覚化される様々なコンテキストに適用され得る。特に、本発明は、キャッシュに格納された１組のウェブページ又は他のウェブオブジェクトによって形成された構造のような、ウェブ関連構造の視覚化に適用され得る。

【０１５７】より一般的には、本発明は、組織図、ファイルシステム階層、ハイパーテキスト階層、ワールドワイドウェブ接続性構造、パーツ分解、ＳＧＭＬ構造、又は任意の他の大きなノード－リンク構造のためのブラウザの提供に適用され得る。このブラウザは、構造又は構造の内容の編集に用いられてもよい。

【０１５８】本発明は、ソフトウェアとしての実施例に関係して述べられてきたが、本発明は専用ハードウェアを用いて実施されてもよい。

【図面の簡単な説明】

【図１】変更を伴うノード－リンク表現のシーケンスを示す模式的なフロー線図である。

【図２】図１の表現シーケンスのような表現シーケンスの表示における一般的な処理を示すフロー図である。

【図３】図１の表現シーケンスのような表現シーケンスを表示する装置の一般的な構成要素を示す模式的な線図である。

【図４】個別の変更を要求する信号に基づいて、図１の＊表現シーケンスのような表現シーケンスを表示する準備における一般的な処理を示すフロー図である。

【図５】システムの模式的な線図である。

【図６】どのように図５のシステムが有向グラフの表現を表示することによってイベントに応答できるかを示すフロー図である。

【図７】図６でどのように最初のレイアウトを実行できるかを示すフロー図である。

【図８】図６でどのように変更されたノード－リンク構造のレイアウトを実行できるかを示すフロー図である。

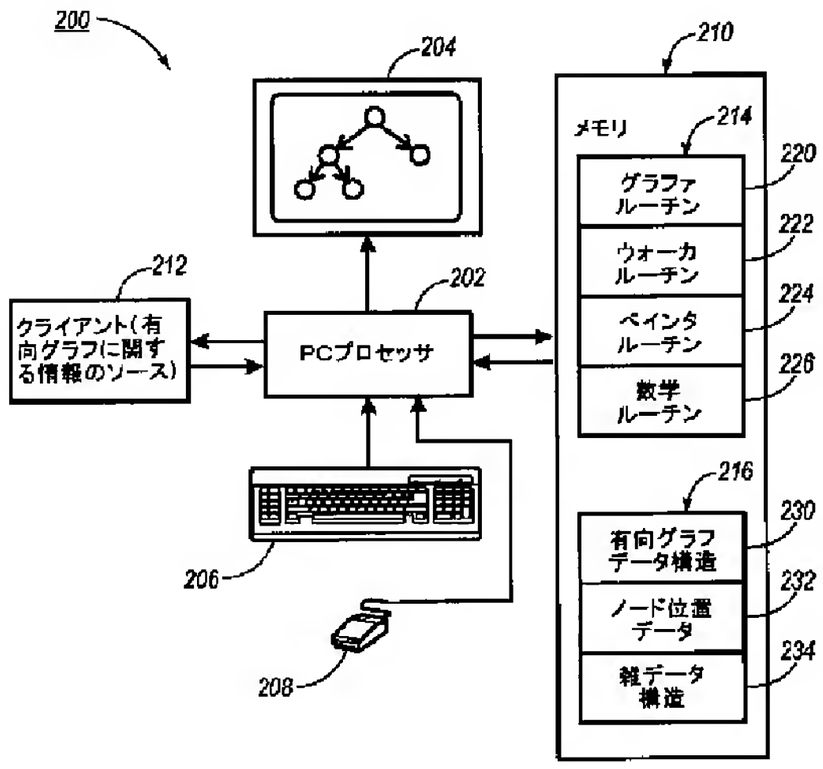
【図９】図８の技術によって生成できる画像の２つのシーケンスを示す図である。

【図１０】図８の技術によって生成できる画像の２つのシーケンスを示す図である。

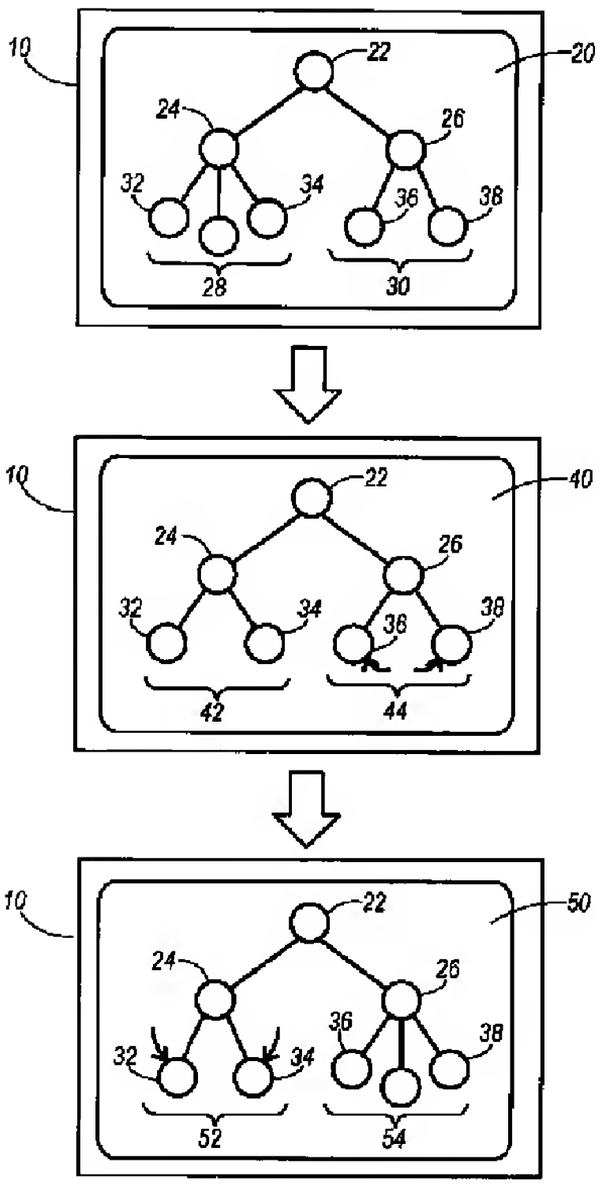
【符号の説明】

１０	ディスプレイ
２０	第１表現
４０	中間表現
５０	最終表現
２１２	クライアント
２２０	グラファルーチン
２２２	ウォーカルーチン
２２４	ペインタルーチン
２２６	数学ルーチン
２３０	有向グラフデータ構造
２３２	ノード位置データ

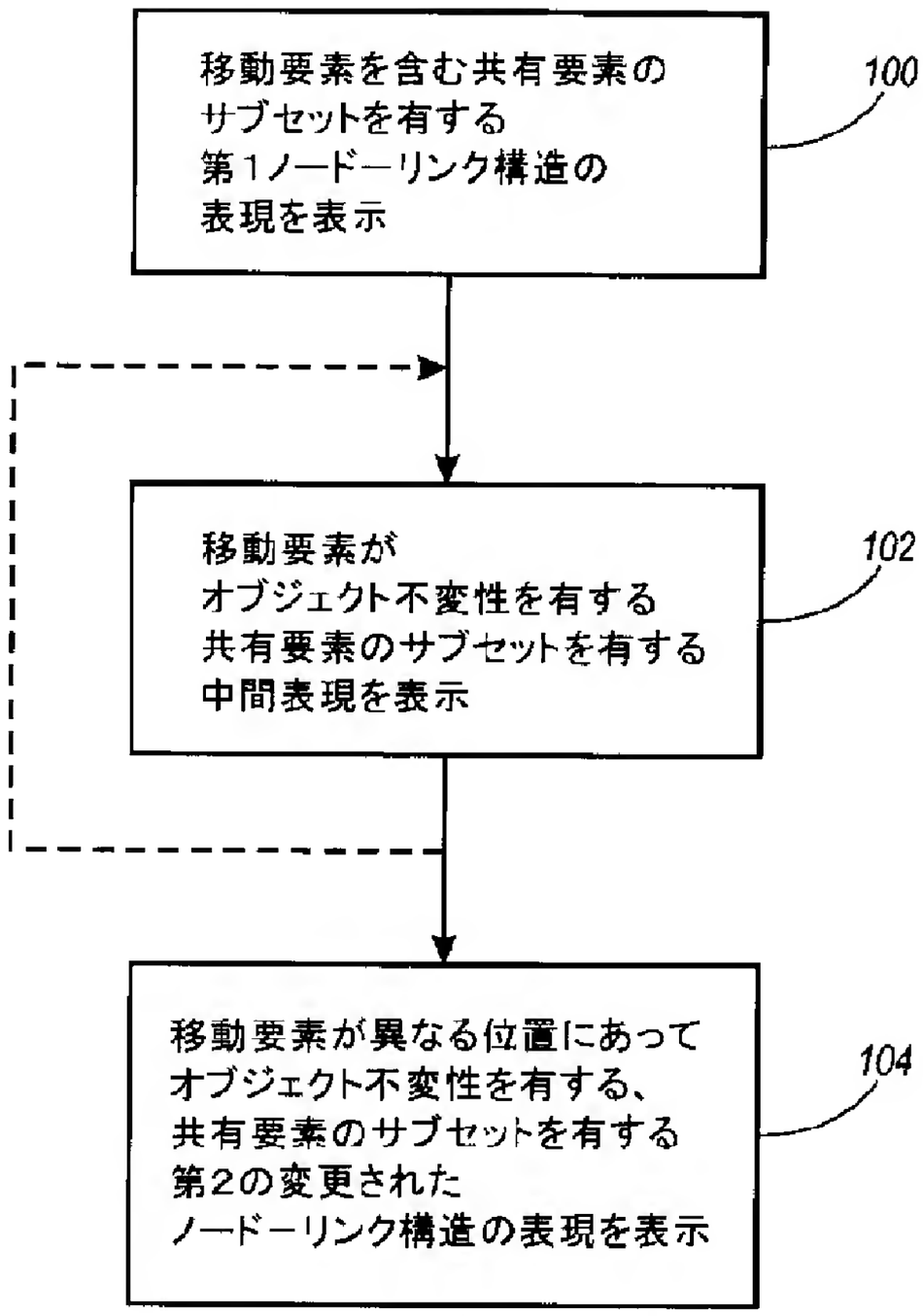
【図５】



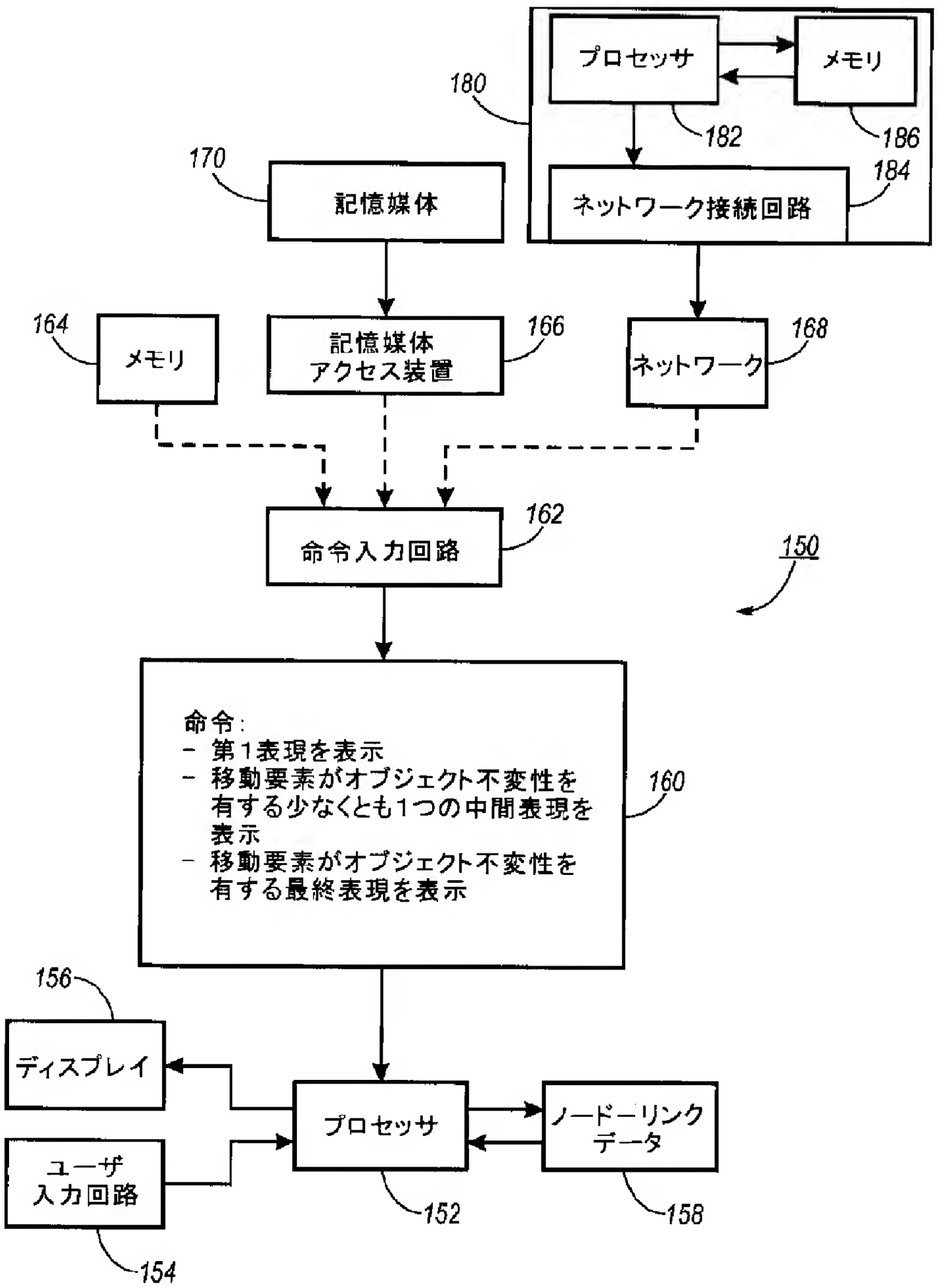
【図１】


10
20
22
24
26
32
34
36
38
28
30
10
40
22
24
26
32
34
36
38
42
44
10
50
22
24
26
32
34
36
38
52
54

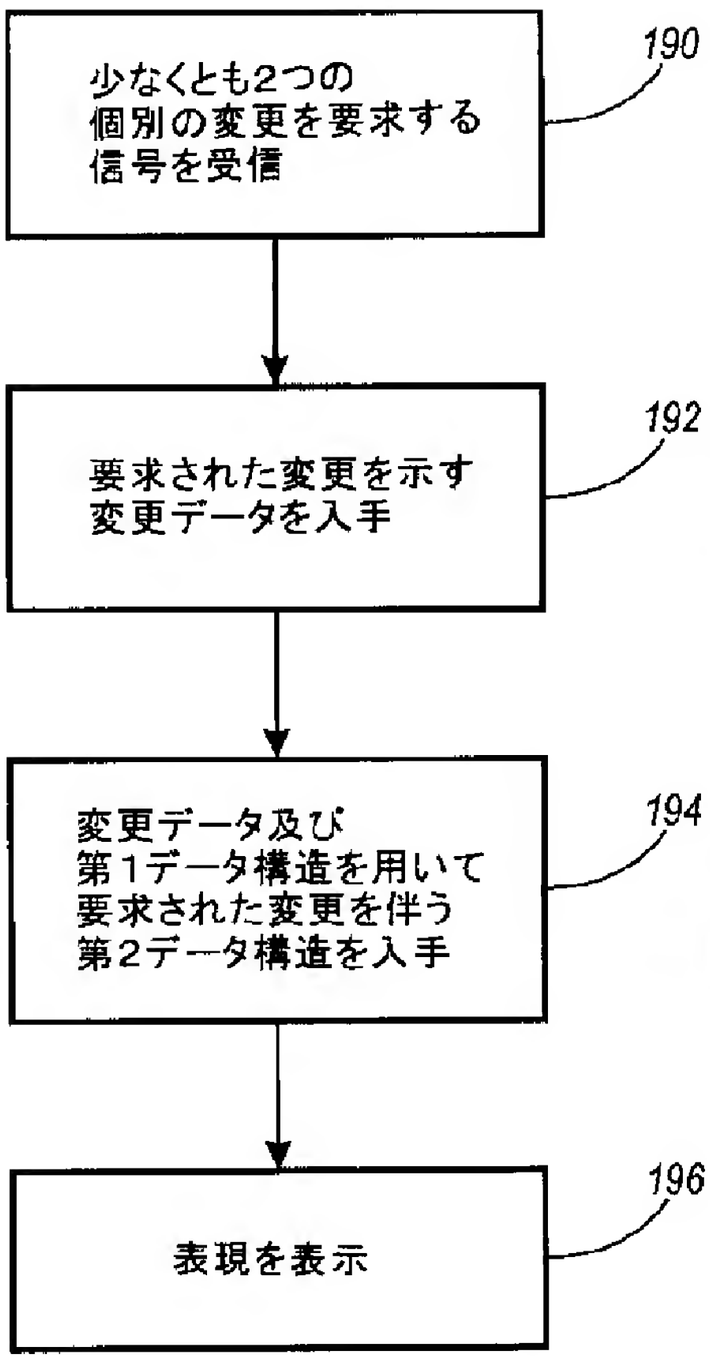
【図２】


移動要素を含む共有要素の
サブセットを有する
第１ノードーリンク構造の
表現を表示
100
移動要素が
オブジェクト不変性を有する
共有要素のサブセットを有する
中間表現を表示
102
移動要素が異なる位置にあって
オブジェクト不変性を有する、
共有要素のサブセットを有する
第２の変更された
ノードーリンク構造の表現を表示
104

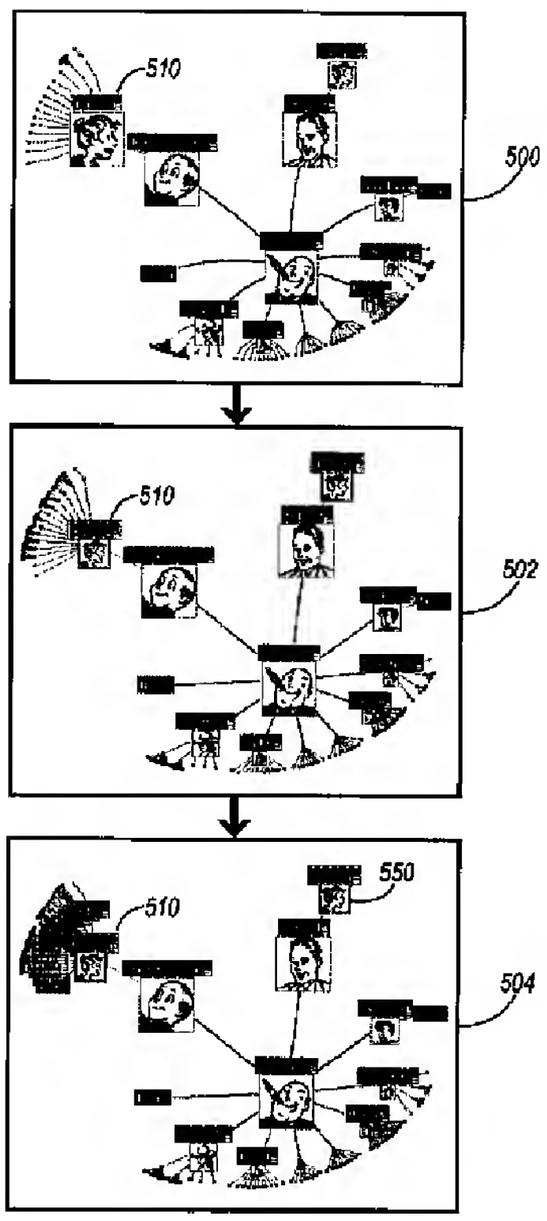
【図３】


プロセッサ
メモリ
180
182
186
170
記憶媒体
ネットワーク接続回路
184
164
メモリ
記憶媒体
アクセス装置
166
ネットワーク
168
162
命令入力回路
150
命令:
- 第1表現を表示
- 移動要素がオブジェクト不変性を有する少なくとも1つの中間表現を表示
- 移動要素がオブジェクト不変性を有する最終表現を表示
160
156
ディスプレイ
プロセッサ
ノードーリンクデータ
ユーザ入力回路
152
158
154

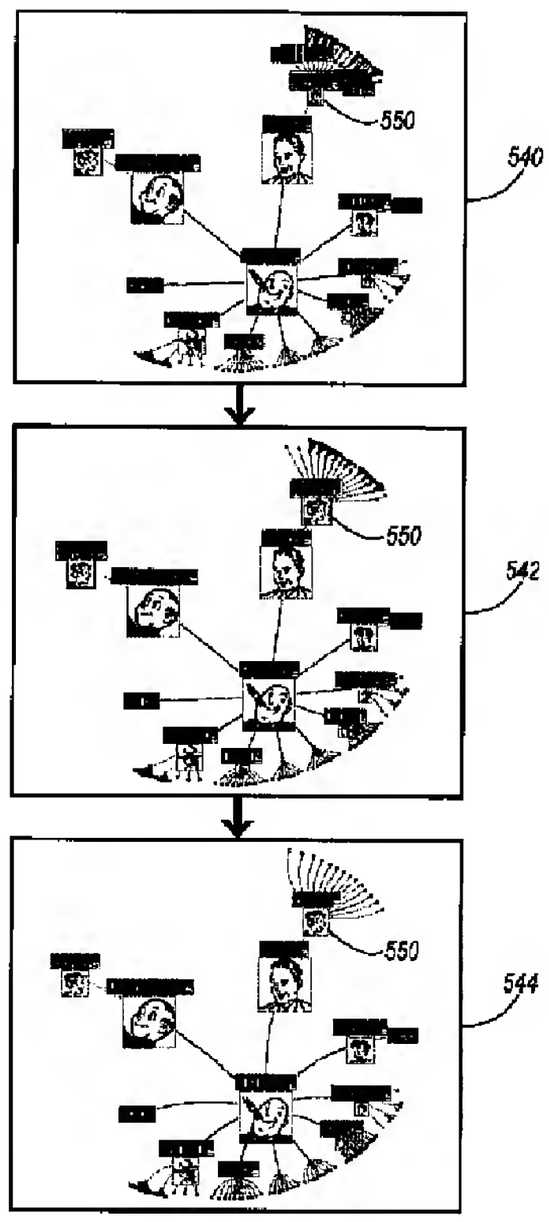
【図４】


少なくとも2つの
個別の変更を要求する
信号を受信
190
要求された変更を示す
変更データを入手
192
変更データ及び
第1データ構造を用いて
要求された変更を伴う
第2データ構造を入手
194
表現を表示
196

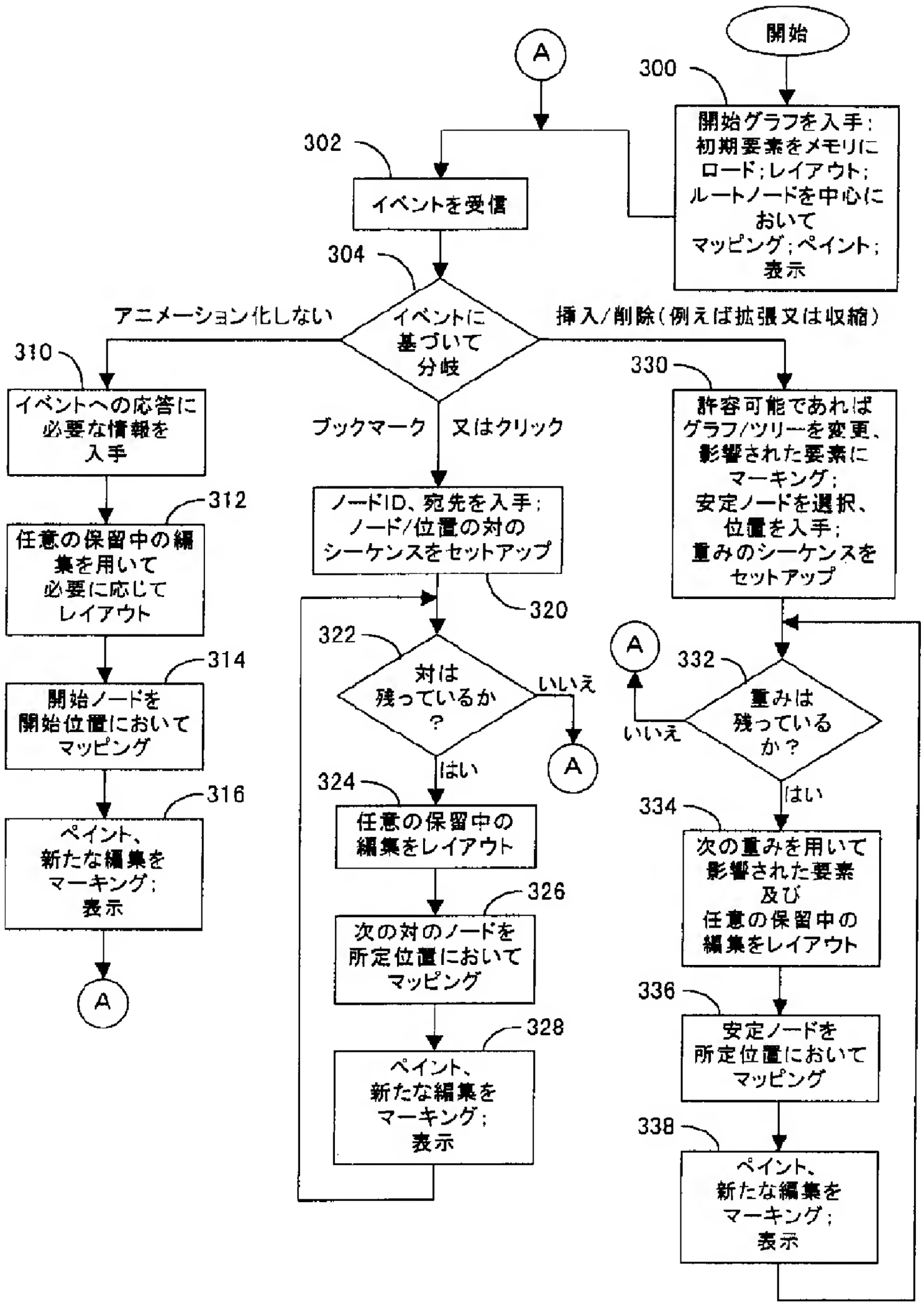
【図９】


510
500
510
502
550
510
504

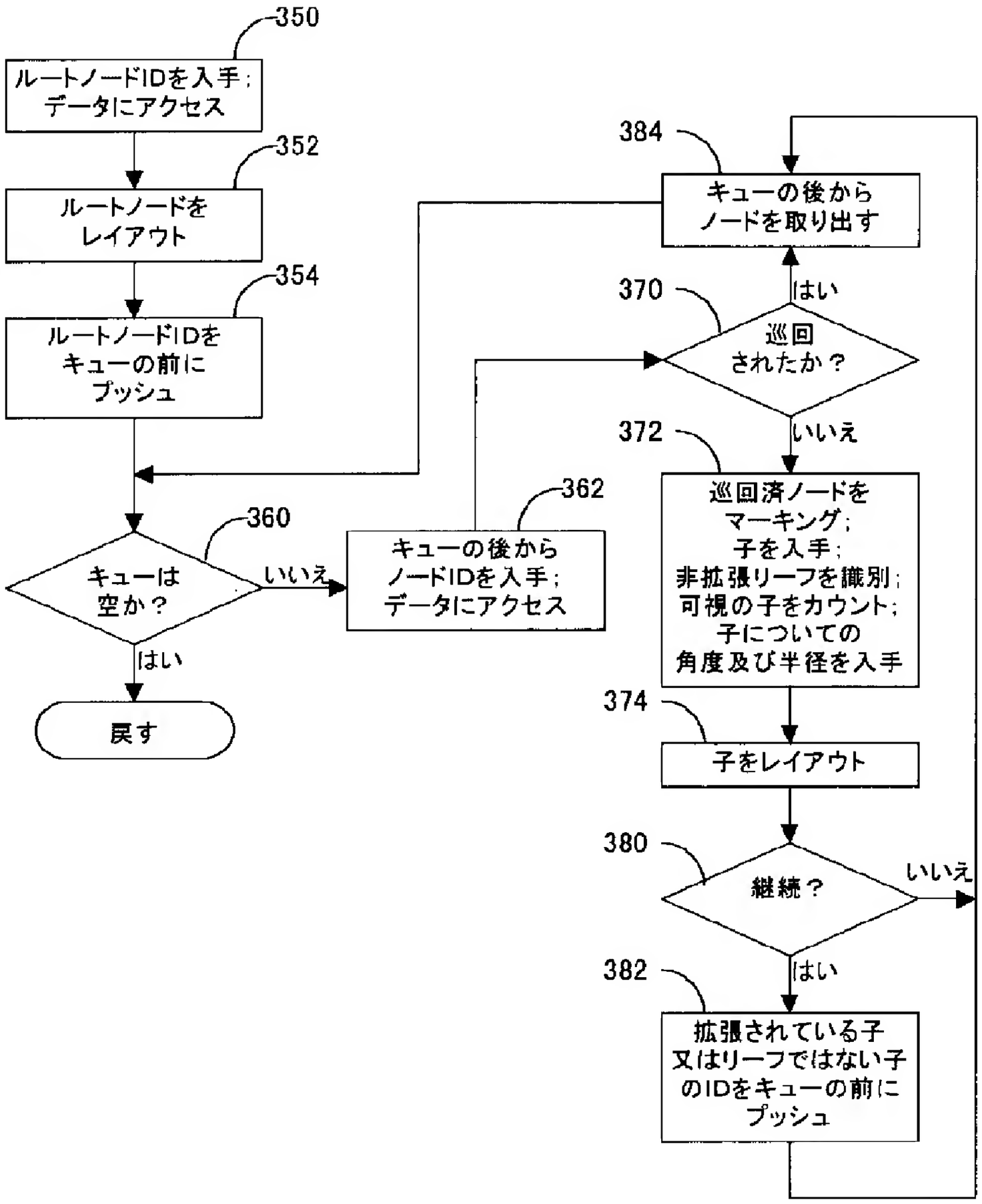
【図１０】


550
540
550
542
550
544

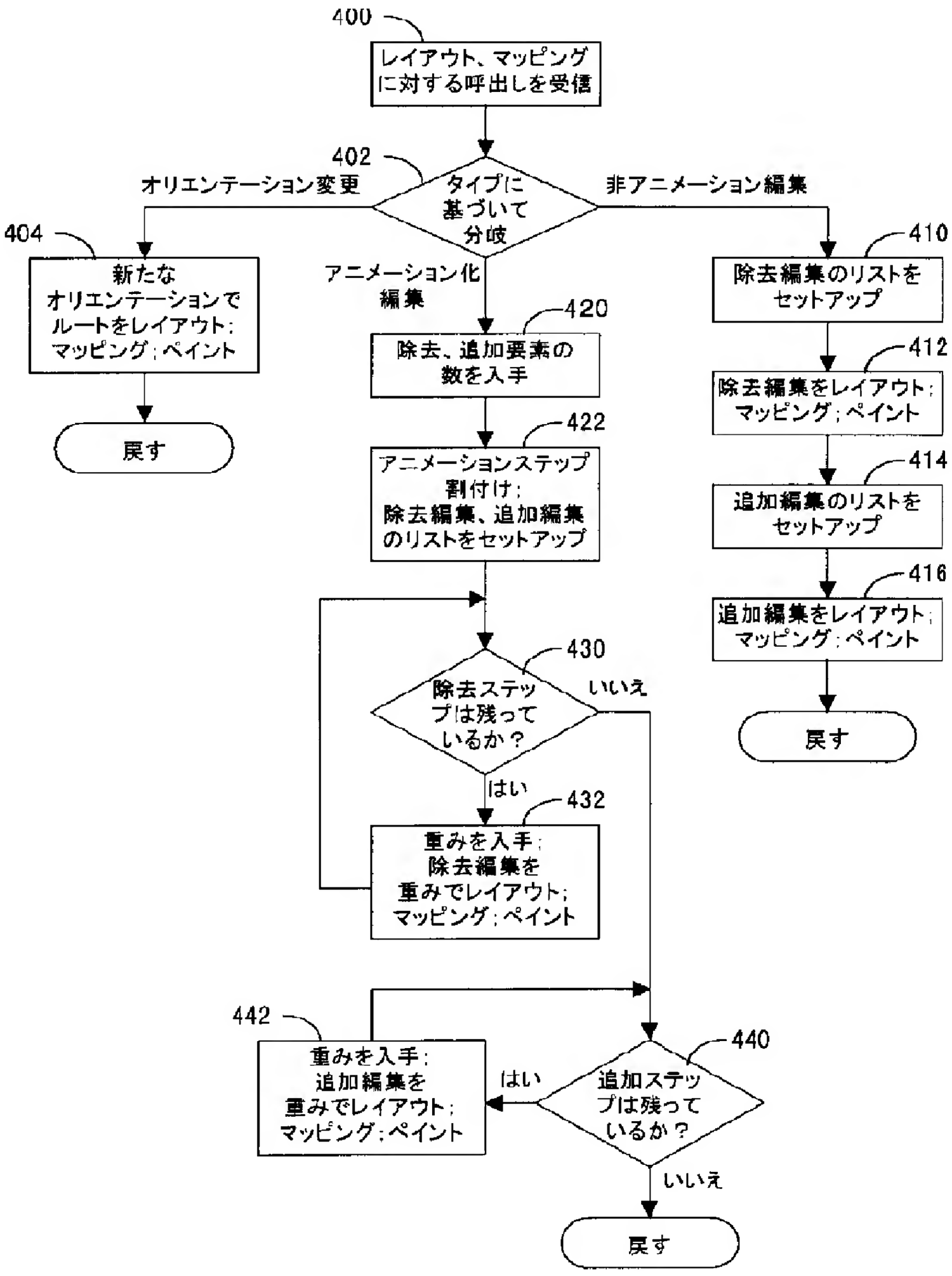
【図６】


開始
300
開始グラフを入手；初期要素をメモリにロード；レイアウト；ルートノードを中心においてマッピング；ペイント；表示
A
302
イベントを受信
304
イベントに基づいて分岐
アニメーション化しない
挿入/削除（例えば拡張又は収縮）
ブックマーク
又はクリック
310
イベントへの応答に必要な情報を入手
312
任意の保留中の編集を用いて必要に応じてレイアウト
314
開始ノードを開始位置においてマッピング
316
ペイント、新たな編集をマーキング；表示
A
ノードID、宛先を入手；ノード/位置の対のシーケンスをセットアップ
320
322
対は残っているか？
いいえ
はい
A
324
任意の保留中の編集をレイアウト
326
次の対のノードを所定位置においてマッピング
328
ペイント、新たな編集をマーキング；表示
330
許容可能であればグラフ/ツリーを変更、影響された要素にマーキング；安定ノードを選択、位置を入手；重みのシーケンスをセットアップ
A
332
重みは残っているか？
いいえ
はい
334
次の重みを用いて影響された要素及び任意の保留中の編集をレイアウト
336
安定ノードを所定位置においてマッピング
338
ペイント、新たな編集をマーキング；表示

【図７】


350
ルートノードIDを入手；
データにアクセス
352
ルートノードを
レイアウト
354
ルートノードIDを
キューの前に
プッシュ
360
キューは
空か？
いいえ
はい
362
キューの後から
ノードIDを入手；
データにアクセス
戻す
384
キューの後から
ノードを取り出す
370
巡回
されたか？
はい
いいえ
372
巡回済ノードを
マーキング；
子を入手；
非拡張リーフを識別；
可視の子をカウント；
子についての
角度及び半径を入手
374
子をレイアウト
380
継続？
いいえ
はい
382
拡張されている子
又はリーフではない子
のIDをキューの前に
プッシュ

【図８】



フロントページの続き

(72)発明者　ラマナ　ビー．ラオ
アメリカ合衆国　94112　カリフォルニア州　サンフランシスコ　イナ　コート　50

(72)発明者　ジョン　オー．ランピング
アメリカ合衆国　94024　カリフォルニア州　ロス　アルトス　エヴァ　アベニュー　1299